

Evaluating the limits of Quantum Approximate Optimization Algorithm parameter transfer at high rounds on sparse Ising models with geometrically local cubic terms

Elijah Pelofske^{1,*}, Marek M. Rams^{2,3}, Andreas Bärtzsch⁴, Piotr Czarnik^{2,3}, Paolo Braccia⁵,
Lukasz Cincio⁵, and Stephan Eidenbenz⁴

¹Information Systems & Modeling, *Los Alamos National Laboratory*, Los Alamos 87545, USA

²Institute of Theoretical Physics, *Jagiellonian University*, Łojasiewicza 11, 30-348 Kraków, Poland

³Mark Kac Complex Systems Research Center, *Jagiellonian University*, Łojasiewicza 11, 30-348 Kraków, Poland

⁴Information Sciences, *Los Alamos National Laboratory*, Los Alamos 87545, USA

⁵Quantum and Condensed Matter Physics, *Los Alamos National Laboratory*, Los Alamos 87545, USA



(Received 20 September 2025; accepted 2 March 2026; published 7 April 2026)

The emergent practical applicability of the Quantum Approximate Optimization Algorithm (QAOA) for approximate combinatorial optimization is a subject of considerable interest. One of the primary limitations of QAOA is the task of finding a set of good parameters, which is usually done using a variational optimization loop. Parameter transfer, or parameter concentration, is a phenomenon where QAOA angles trained on problem instances that are self-similar tend to perform well for other problem instances from that similar class. This suggests a potentially highly efficient and scalable nonvariational learning method for QAOA angle finding. In this work, we systematically study QAOA parameter transferability from small problem sizes (16 and 27 decision variables) onto large problem instances (up to 156 qubits) for heavy-hex graph Ising models with geometrically local higher-order terms using the Julia based QAOA simulation tool *JuliaQAOA* to perform classical angle finding for up to 49 QAOA layers (p). Parameter transfer of the fixed angles is validated using a combination of full state vector, projected entangled pair states, matrix product state, and LOWESA numerical simulations. We find that the QAOA parameter transfer from single instances applied to other (unseen) problem instances does not in general provide monotonically improving performance as a function of p —there are many cases where the performance temporarily decreases as a function of p —but despite this the transferred angles have a general trend of improved expectation value as the QAOA depth increases, in many cases converging close to the true ground-state energy of the 100+ qubit instances. We also sample the hardware-compatible Ising models using the ensemble of transfer-learned QAOA parameters on several superconducting qubit IBM quantum processors with 127, 133, and 156 qubits. We find continuous solution quality improvement of the hardware-compatible QAOA circuits run on the IBM noisy intermediate-scale quantum processors up to $p = 5$ on *ibm_fez*, up to $p = 9$ on *ibm_torino*, and up to $p = 10$ on *ibm_pittsburgh*.

DOI: [10.1103/p2lg-z4kn](https://doi.org/10.1103/p2lg-z4kn)

I. INTRODUCTION

The Quantum Approximate Optimization Algorithm (QAOA) [1–5] is designed to sample good solutions of combinatorial optimization problems. QAOA is relatively noisy intermediate-scale quantum (NISQ) [6] computer friendly in that it is feasible to implement the algorithm at the scale of near-term noisy quantum processors (short depth, small qubit counts). There have been a considerable number of studies on the properties of QAOA using both numerical simulations and current quantum processors, in particular in comparison to existing classical algorithms that solve combinatorial

optimization problems [7–14]. One of the central challenges with the algorithmic study of QAOA is scaling sampling analysis with respect to number of variables and QAOA depth. This scaling requires either somewhat small-scale numerical simulations (limited to tens of qubits for exact state vector computations, or up to hundreds of qubits using tensor network methods or similar types of approximate numerical techniques) or quantum hardware simulations that are subject to a variety of sources of error. QAOA requires hyperparameter selection that in turn generally requires significant compute time, especially if this process is performed on the quantum device. However, the QAOA parameter learning is distinct from many other types of variational quantum algorithms [15] because the number of trainable parameters does not scale, per layer, with the number of variables in the problem, meaning that parameter consistency among problems with varying problem sizes, or coefficient distributions, can be examined. It turns out that QAOA parameters do indeed exhibit parameter consistency for problem instances that share some similarity, such as similar connection structure. This property has been

*Contact author: epelofske@lanl.gov

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

studied and utilized for efficient QAOA parameter finding in numerous studies [16–36]. Many of these types of parameter transfer studies are able to condense QAOA angle usage into fixed angle formulas, at least for average cases of certain problem types and typically at fairly low depth [37]. If QAOA can be used in a nonvariational learning manner, then the prospects of using QAOA for meaningfully sized problem instances on large-scale error-corrected quantum computers are good—the variational learning overhead of QAOA can be quite significant. Parameter setting heuristics, namely, from parameter concentration, will enable this approach to use QAOA at a much larger scale [38]. There do exist other nonvariational QAOA approaches, such as using linear-ramp adiabatic schedules [39,40], which at sufficiently large p guarantee optimality.

Here, we show through extensive numerical simulation some of the favorable characteristics of parameter transfer, and some of the limitations. More precisely, we use QAOA parameters trained on small, state vector simulation-addressable problem instances to execute full-scale QAOA runs on several IBM quantum processors. Finally, we validate the QAOA angle performance using extensive tensor network simulations, in particular projected entangled pair states (PEPS). The QAOA circuits and optimization problems that we use in this study are carefully designed to be extremely hardware compatible to the heavy-hex graph and are therefore very NISQ computer friendly (see Refs. [41,42]). The sparsity of the variable connections in these optimization problem instances makes them an excellent test case for studying fundamental algorithmic engineering questions about QAOA. In this case, we are focusing on high QAOA depth parameter transfer.

Previous studies have performed NISQ-hardware simulations of relatively small-scale optimization problems using approximate digital quantum optimization algorithms. Reference [43] used variational digitized counterdiabatic quantum optimization to sample problems with higher-order terms, on a 156-qubit IBM quantum processor. References [41,42] used the Quantum Approximate Optimization Algorithm to sample 127 variable problems with higher-order terms on a 127-qubit IBM superconducting processor—and Ref. [28] then extended this up to a 414-qubit IBM quantum processing unit (QPU). Reference [10] simulated QAOA for sampling MAX-3-SAT problems for up to 22 variables up to depth $p = 2$, and 10 variables for up to depth $p = 20$ using trapped ion quantum processors. Reference [44] sampled maximum cut on random three-regular graphs for up to $p = 2$ using the IBM QPUs. Reference [11] sampled up to 32 variable maximum cut problems on random three-regular graphs, for up to $p = 11$, using a trapped ion quantum processor. In regard specifically to applying QAOA using parameter transfer, the largest scale previous study was Ref. [28], which was able to provide high-quality angles for up to QAOA depth 5 on 127-qubit sparse Ising model instances—in this study, we go well beyond that up to depth 49 and 156-qubit instances, using a broader ensemble of QAOA angles.

In this study, the high-depth scaling of QAOA parameters is characterized using an ensemble of learned QAOA angles on small (e.g., less than 30 qubits) problem instances whose parameters are then transferred to be used on much larger problem instances (up to 156 qubits). The central question

that is examined is how far parameter transfer QAOA angles can be used, with respect to p . This is an important question because although parameter transfer seems to be incredibly successful for finding good performing QAOA angles, there are limits to direct parameter training for parameter transfer—namely, for the task of transferring parameters up to larger problem sizes, better angles cannot be learned past an approximation ratio of 1, and higher QAOA depths are generally required as problem sizes increase. It is not clear how well parameter transfer can be applied to various combinatorial optimization problem instances in general, but it is crucial to understand where transferred QAOA angles fail to perform, or perhaps work well in certain cases. Remarkably, one of the findings of this study is that the intensely optimized QAOA angles, using the classical numerical state vector simulation tool *JuliaQAOA*, recover the smooth-schedule properties of discretized quantum annealing (adiabatic)-type protocols.

Using an ensemble of numerically trained small problem instance QAOA angles allows very large QAOA circuits to be executed on 127-, 133-, and 156-qubit IBM quantum processors, using no on-device variational learning. These QAOA executions show clear signal of improved sampling at higher-depth circuits, in particular having a consistently improving approximation ratio starting at $p = 1$, up to $p = 9$ on *ibm_torino*, up to $p = 10$ on *ibm_pittsburgh*, up to $p = 7$ on both *ibm_aachen* and *ibm_kingston*, and up to $p = 3$ on *ibm_kyiv* (larger depths of continued improvement are limited by hardware noise on these devices). This is the largest reported continual QAOA performance improvement, as depth increases, on a quantum computer to date. Moreover, this study demonstrates, primarily by performance validation using tensor network approximation methods, that parameter transfer is a highly efficient technique for high-depth QAOA circuits to be executed, with no variational learning.

II. METHODS

Here, we first briefly summarize QAOA. Given a combinatorial optimization problem defined by a cost function $C(z)$ on n -spin states $z \in \{+1, -1\}^n$, QAOA is defined by the following components:

- (1) an initial state $|\psi\rangle$, in this study we use the standard $|\psi\rangle := |+\rangle^n$,
- (2) a cost Hamiltonian H_C that represents the cost values, here $H_C := \sum_z C(z)|z\rangle\langle z|$;
- (3) a phase separating Hamiltonian H_P that is used to phase basis states z by a parametrized function of their cost value, here $H_P := H_C$; and
- (4) a mixing Hamiltonian H_M , which is used to give parametrized interference between basis states; here, we use the standard transverse field mixer $H_M = \sum_{j=1}^n X_j$.

QAOA then prepares a parametrized quantum state

$$|\vec{\beta}, \vec{\gamma}\rangle = e^{-i\beta_p H_M} e^{-i\gamma_p H_P} \dots e^{-i\beta_1 H_M} e^{-i\gamma_1 H_P} |\psi\rangle, \quad (1)$$

where we have the following parameters:

- (1) an integer $p \geq 1$ that describes the number of rounds [45] to run the algorithm;
- (2) a vector of reals $\vec{\beta} = (\beta_1, \dots, \beta_p)$ of length p , which parametrizes the layer-wise time evolution of H_M ; and

(3) a vector of reals $\vec{\gamma} = (\gamma_1, \dots, \gamma_p)$ of length p , which parametrizes the layer-wise time evolution of H_p .

Measuring the state $|\vec{\beta}, \vec{\gamma}\rangle$ in the computational basis returns a sample solution z of cost value $C(z)$ with probability $|\langle z | \vec{\beta}, \vec{\gamma} \rangle|^2$, with a mean expectation value of $\langle \vec{\beta}, \vec{\gamma} | H_C | \vec{\beta}, \vec{\gamma} \rangle$. Combined, the real-valued parameters are usually called *angles*—and these are the parameters that are typically tuned using a black-box optimizer algorithm run on a classical processor. Typically, these parameters are tuned such that the mean expectation value is optimized, which is the training performed in the pretraining stage described next.

The QAOA angle finding in this study is carried out using the Julia-based software package [46] `JulIQAOA` [47], where the angle finding is performed using basin hopping optimization, which in turn uses gradient descent, and all angle optimization begins with uniformly random guesses at $p = 1$. `JulIQAOA` is an efficient numerical simulation angle finding tool that can heuristically generate high-quality QAOA angles that are not necessarily globally optimal, given a specific set of $C(z)$ cost values for all feasible solutions. As far as we are aware, finding globally optimal QAOA parameters is a very hard computational problem; we are not aware of tools or techniques that would allow us to verify global optimality of QAOA parameters (e.g., up to some precision), even for relatively small optimization problem instance sizes. `JulIQAOA` has been shown to be a highly effective QAOA angle learning tool [10,13,48,49]. One basin hopping iteration is used for each QAOA round, and p is incremented in step sizes of 1. *Angle extrapolation* is also used within each `JulIQAOA` p angle optimization pass—where $p + 1$ angles are initialized using the best angles found at the preceding p step. Note that similar ideas have been used in previous studies to compute good QAOA angles at $p > 1$ [50]. At each step of p past $p = 1$, the angle learning is repeated until the mean expectation value has improved over the previous step of p —each additional angle learning step (e.g., at $p + 1$) uses random angle initialization and since the learning is not necessarily optimal, different initializations can result in different learned QAOA angle quality. Similar sets of heuristic rules used in `JulIQAOA` have been developed in other studies, motivated by similar ideas of iterative or recursive exploration of the $p + 1$ energy landscape using the information gained from the p energy landscape [50–53]. Importantly, this `JulIQAOA` simulation is performing full state vector simulation, meaning that the training procedure has at least one advantage of not being hindered by shot noise (e.g., finite sampling effects). However, because of the size of the systems, in particular the 27-qubit Ising models, this simulation is quite memory and compute time intensive. The angle training used both high performance computing resources and many weeks of elapsed wall clock time (including reattempts at optimization where the expectation value does not improve monotonically). The aim of this study is to make use of this precomputation of the QAOA angles at small system sizes, where such state-vector-based training is at least tractable, and apply these parameters to much larger QAOA circuits. We have found that this particular QAOA angle learning tool is the most effective tool that we are aware of; other optimization techniques, such as gradient-free optimization methods, fail systematically at high QAOA depths.

The combinatorial optimization problems are all Ising models, which makes them unconstrained minimization problems. This class of Ising models was introduced as a very hardware-compatible Ising model for the IBM quantum hardware heavy-hex graph structure [54–56] by Refs. [41,42]. These Ising models include geometrically local cubic terms (ZZZ terms) that make the problems computationally harder, but the phase separating Hamiltonian circuit implementation is highly optimized to have strictly a two-qubit gate depth of 6 per p (notably, using no additional two-qubit gates to handle the higherorder term compared to the heavy-hex quadratic interactions) [41,42]. This hardware compatibility makes these circuits incredibly NISQ computer friendly, and both these Ising models and QAOA circuits to sample these Ising models have also been used in several prior studies [28,57,58]. Importantly, these Ising models are designed with the aim of being NISQ-friendly test cases to evaluate QAOA properties, namely, for running on near-term noisy (heavy-hex graph) IBM quantum processors, not with the aim of being extremely computationally challenging. These Ising models, primarily because of the sparsity of spin interactions, lack of long-range interactions, and because of the relatively small number of variables used (e.g., in this study, the largest problem instance we consider is 156 decision variables), can be solved with classical optimizers such as simulated annealing or provable optimality solvers such as Gurobi extremely quickly (on the order of seconds or faster) [28,41,42,59]. This class of optimization problems is defined by the following cost function $C(z)$ that simultaneously can be understood as a classical Hamiltonian,

$$H_C = \sum_i d_i Z_i + \sum_{(i,j)} d_{i,j}^{(2)} Z_i Z_j + \sum_i d_{i,j,k}^{(3)} Z_i Z_j Z_k. \quad (2)$$

The heavy-hex graph $G = (V, E)$ is defined by a set of vertices V and edges E . The Hamiltonian is defined by single-spin terms indexed by $i \in V$, the spin-spin interactions are indexed by the edges $(i, j) \in E$, and finally the local cubic terms are indexed by (i, j, k) . The cost function takes as input a vector of spins $z = (z_1, z_2, \dots, z_n) \in \{+1, -1\}^n$. Every heavy-hex lattice is bipartite and the bipartition can be defined as $V = V_2 \sqcup V_3$ with $E \subset V_2 \times V_3$, where V_3 consists of vertices of maximum degree 3 and V_2 consists of vertices of maximum degree 2. W is the set of vertices in V_2 where each node has degree 2, and the spins in this set W define the center spin of each of the geometrically local cubic terms in Eq. (2), indexed as (i, j, k) . All variable states are spins, where the states are either $\{+1, -1\}$, and the polynomial coefficients, $d_i, d^{(2)}, d^{(3)}$, are randomly (and independently) chosen to be either $+1$ or -1 , making these a type of hardware-compatible disordered $\pm J$ models. The complexity class, for exact optimal solution computation, that the Ising models of the form Eq. (2) exist within is not currently known [59]. The QAOA circuits use a direct spin to qubit mapping; each qubit represents a single spin in the logical Ising model.

Figure 13 in Appendix A gives a circuit diagram translation from the hardware-compatible Ising model to the corresponding QAOA circuit diagram, where the most expensive component of the circuit in terms of gate complexity is the phase separator. The QAOA circuit simulations run on the

TABLE I. Summary of the 12 IBM quantum processing units used in our QAOA hardware experiments. The given basis gates are the gates that the QAOA circuits are compiled to (locally) before executing on the device—in some cases, such as on `ibm_marrakesh`, there are now additional gates available, which were not used for the QAOA circuit compilation in this study.

Processor	No. of qubits	No. of two-qubit gates	Basis gates	QPU names
Eagle r3	127	144	ECR, ID, RZ, SX, X	<code>ibm_sherbrooke</code> , <code>ibm_kyiv</code> , <code>ibm_brisbane</code> , <code>ibm_kyoto</code> , <code>ibm_nazca</code> , <code>ibm_brussels</code> , <code>ibm_strasbourg</code>
Heron r1	133	150	CZ, ID, RZ, SX, X	<code>ibm_torino</code>
Heron r2	156	176	CZ, ID, RZ, SX, X	<code>ibm_fez</code> , <code>ibm_marrakesh</code> , <code>ibm_kingston</code> , <code>ibm_aachen</code>
Heron r3	156	176	CZ, ID, RZ, SX, X	<code>ibm_pittsburgh</code>

IBM hardware do not use any dynamical decoupling, and do not use any type of quantum error mitigation. The QAOA circuits rely on computing a 3-edge-coloring of the heavy-hex hardware graph—the 3-edge-coloring used in all of the hardware simulations is randomly generated using minimum edge coloring heuristic algorithms from NetworkX [60]. That fixed edge coloring is then used in all p layers (this is recalculated for each Ising model instance). In principle, these edge colorings could be randomized to generate circuit invariants with slightly different noise profiles when executed on the hardware, but we leave this for future study. Additionally, these edge colorings could be optimized to have distinct (average) gate times in each layer, thus reducing the overall circuit time [61]. During compilation testing, we did attempt this type of 3-edge-coloring partitioning. However, in practice, the two-qubit gate times of the IBM processors used in this study have very nearly uniform times, which means that no significant reduction of circuit execution time can be gained, and randomization of the two qubit 3-edge-coloring does not yield variance in total circuit durations. In principle, if a quantum computer had two-qubit gate timings that were highly variable, an optimized staggered edge coloring (e.g., three colors that correspond to short, median, and long gate times) could be used to compile circuits such that the total circuit duration was minimized. In other words, if the two-qubit gate timings effectively replicated a 3-edge-coloring via distinct timing bins, then compiling to that edge coloring could yield substantial improvements in total circuit duration.

All of the quantum hardware simulations are performed using IBM quantum programmable fixed-frequency superconducting transmon qubit [62] devices, which are summarized in Table I. These devices have the heavy-hex two-qubit connectivity graph [54]. The quantum circuits are defined and executed using the Qiskit Python 3 library [63,64]. The circuits are transpiled using a circuit optimization level of 3 before being sent to the quantum computer. Qiskit versions 1.2.2, 0.45.0, and 2.1.2 were used across all of the circuit executions on the IBM QPUs. The circuits are run on the IBM quantum computer in the form of discrete *jobs*—for all devices, the shot count per job is 1000 000, with the exception of `ibm_torino` where because of device usage limitations only 5000 shots were run per job (thus requiring more jobs to be run to get an equivalent total shot count to the other devices). One of the consequences of this is higher QPU compute time used in total for `ibm_torino` because there is a time cost associated with running many more individual jobs. Occasionally, for some of the IBM QPUs a job would fail to execute due to an unknown internal error—in these cases, the

job was simply resubmitted to the backend (failed jobs are not counted toward reported aggregate statistics such as QPU time).

One way to quantify the QAOA solution quality, besides the energy of the Ising model Hamiltonian, is the approximation ratio, which given a sample energy e is defined as

$$\text{approximation ratio} = \frac{C_{\text{Max}} - C_e}{C_{\text{Max}} - C_{\text{Min}}} \quad (3)$$

for the Ising model where C_{Max} is the global maximum possible energy and C_{Min} is the global minimum possible energy. C_e is the cost function $C(z)$ [Eq. (2)], also called the energy, of a single spin configuration. This approximation ratio formula is maximized for minimum energies and minimized for maximum energies, meaning that this is intended to be applied to minimization problems.

All binary solutions obtained by the various solvers used in this study are adapted to spins (variable states of $\{+1, -1\}$) via the mapping of $1 \mapsto 1$, $0 \mapsto -1$. Optimal solutions of the Ising models considered in this study, specifically the optimal energy (both minimum and maximum cost function value), are deterministically computed using the commercial mathematical optimization software packages CPLEX and Gurobi [65,66]. The random J coupling of these models means that these can be viewed as a type of sparse spin glass model (e.g., a heavy-hex spin glass model), in particular the $\pm J$ distribution leads to some competing frustration throughout the model, and therefore fairly high ground-state (GS) degeneracy. However, we do not investigate any degeneracy properties, such as degeneracy lifting, or frustration properties, of these models in this study.

A. Projected entangled pair states simulation methods

The scale of the quantum simulations here means that it is intractable to perform full state-vector simulations. However, there exist approximation tensor network-based methods that can in many instances perform high-quality simulations of these types of quantum circuits, namely, on the order of hundreds of qubits and with entangling graph structure (in this case defined by the IBM heavy-hex hardware graphs), which are relatively sparse. To this end, it would be useful to validate how well these QAOA angles work on very large instances, using simulations that have lower noise than the IBM QPUs themselves. For this, we turn to PEPS simulations, complemented by matrix product state (MPS) and LOWESA simulations at low depth to cross-verify that these simulations agree. Tensor network-based approximation of

low-entanglement digital quantum circuits is a versatile field [67–69] with a number of different tools and allows us in this case to verify, with some reasonable level of numerical approximation, the expected performance of QAOA on this large set of parameter-transferred angles.

Appendix D compares different tensor network contraction methods, and we find that belief propagation (BP) offers a very good convergence of local expectation values, while being more efficient than alternative methods. Therefore, we use BP-based simple update [70,71] (which is equivalent to the original simple update algorithm [72]) to simulate the time evolution of the PEPS representation of the quantum system, and BP to perform the final contraction of the tensor network for calculation of the expectation values or sampling (unless specifically stated otherwise). This is corroborated by Refs. [73–76] that found similar properties of simple update and BP specifically for digital circuits defined on the heavy-hex connectivity graph.

Our PEPS simulations up to bond dimension of $D = 64$, including sampling, have been performed using the YASTN tensor network package [77], and at $D = 128$ were performed using a slightly different in-house implementation built on top of ITensors [78,79]. The YASTN-based QAOA circuit simulation implementation is publicly available on Github [80]. The difference between these two implementations is in how the geometrically local cubic terms are handled, resulting in a trade-off between computational cost and accuracy at any fixed bond dimension value. In the YASTN-based implementation, we assign each PEPS tensor to a group of few qubits, so that each cubic term results in an update of two nearest-neighbor PEPS tensors. In the in-house implementation, each PEPS tensor corresponds to a single qubit and each cubic term results in an update of three PEPS tensors. Thanks to the grouping of spins, the YASTN-based implementation approximates the exact evolution of the PEPS state more accurately for a given bond dimension D . However, it incurs an extra numerical cost due to having to deal with a larger physical space. In contrast, the in-house implementation compensates for its lower accuracy at fixed D by allowing for larger values of the latter, thanks to its lighter computational cost.

B. Matrix product state simulation methods

There exist other types of approximate simulation methods that have been able to approximate larger scale quantum circuits, specifically in the case of heavy-hex connectivity circuits, namely, LOWESA [81,82] and MPS [73]. For this study, we focus on using PEPS simulations because PEPS is particularly suited for 2D system simulations. However, we do evaluate how well MPS and LOWESA perform on simulating these QAOA circuits for two purposes. First, at low circuit depth, we see very good agreement between all three methods, thus validating the accuracy of all three simulators in the low-entanglement/low- p regime. However, at higher circuit depths, we see breakdown of numerical stability and convergence, in particular for LOWESA, thus highlighting the necessity of validating the QAOA simulations with PEPS. Appendix C details these LOWESA simulations.

Here, we detail the one-dimensional tensor network simulations that we perform on these QAOA circuits, in order

to additionally validate the performance of the PEPS simulations. In particular, we expect that if the circuits generate sufficiently low entanglement, we should observe very good agreement between PEPS and MPS at sufficiently large bond dimensions.

Here, we represent the state $|\vec{\beta}, \vec{\gamma}\rangle$ in Eq. (1) with MPS. The action of $e^{-i\beta_j H_M}$ and $e^{-i\gamma_j H_C}$ is simulated with a version of time-evolving block decimation [67]. MPS uses a refinement parameter (called bond dimension and denoted by χ here) that controls the accuracy of the simulation. The Hamiltonian H_C in Eq. (2) contains a significant number of nonlocal (from MPS point of view) interactions and therefore accurate description of $|\vec{\beta}, \vec{\gamma}\rangle$ may require large bond dimension.

Our simulation algorithm collects the terms in H_C acting on the same qubits and decomposes $e^{-i\gamma_j H_C}$ into a product of three-qubit unitaries. It turns out that those unitaries can be exactly represented by a matrix product operator with bond dimension 2. It can be done by performing singular value decomposition of a properly reshaped unitary [83]. This way of treating nonlocal interactions improves the efficiency of our MPS simulation algorithm. Our algorithm collects terms in the Hamiltonian H_C into layers of nonoverlapping unitaries. Unitaries $e^{-i\gamma_{Z_i Z_j Z_k}}$ and $e^{-i\gamma_{Z_{i'} Z_{j'} Z_{k'}}}$ belong to the same layer if $i \geq k'$ or $i' \geq k$. MPS is compressed after all unitary gates in a given layer are applied, possibly resulting in a loss of fidelity. The compression errors (expressed as the sum of discarded eigenvalues of the density matrix of the half of the system) are tracked and found to decay rapidly with χ . The application of $e^{-i\beta_j H_M}$ to an MPS is trivial and does not require compression.

III. RESULTS

The results are organized into subsections as follows. First, in Sec. III A we present the trained QAOA angles, which we will then use for the remainder of the study as fixed QAOA parameters and evaluate how well they work for direct parameter transfer on much larger problem sizes. Next, Sec. III B validates how well some of these fixed parameters work when transferred onto new (unseen) test-problem instances, using exact classical state vector simulations (up to 27 qubits).

Section III C presents extensive large-scale IBM NISQ computer QAOA executions, using the ensemble of transferred QAOA angles. This section includes extensive QAOA sampling result analysis, including reporting large QAOA volume measures.

Section III D presents extensive classical tensor network simulations, specifically PEPS, of the high-depth QAOA circuits, validating how well the QAOA parameter transfer works on Ising model problem sizes of up to 156 spins. This includes ground-state sampling rates and convergence analysis of the PEPS simulations. Complementing the PEPS simulations, Sec. III E then further validates PEPS simulation accuracy using MPS tensor network simulations. Combined, Secs. III D and III E (as well as Appendix C) show strong numerical evidence that large-scale parameter transfer QAOA would work, up to high-depth p QAOA, on quantum computers with significantly lower error than the actual quantum computer hardware results in Sec. III C.

A. Classical QAOA angle learning with state vector simulations

Figure 1 plots the numerical quantities of the learned QAOA angles β and γ for the ensemble of 12 different 16-qubit heavy-hex spin glass instances, learned until $p = 49$ or the mean (noiseless) energy is at least 0.01 (energy) below the optimal ground-state energy. During training, the QAOA angles are not restricted or bounded to a certain range. Figure 2 plots the same but for the ensemble of 27-qubit instances. This set of 24 QAOA angles are the parameters that are used in the entirety of this study for evaluating how well, or not, parameter transfer works up to these very high-depth QAOA circuits.

Our trained angles all lie in the range $[0, 2\pi)$; however, there are several periodicities and symmetries that enable us to further transfer the angles into mathematical equivalent angles plotted in Figs. 1 and 2:

(1) The phase separator $e^{-i\gamma H_p}$ is π -periodic in γ , since for each instance defined in Eq. (2), $H_p = H_C$ has only odd or only even integer eigenvalues;

(2) the mixer $e^{-i\beta H_M}$ is π -periodic up to a global phase, as $e^{-i(\pi+\beta)X} = e^{-i\pi X} e^{-i\beta X} = -Id e^{-i\beta X}$; and

(3) H_C and H_M are real valued and satisfy time reversal symmetry, hence $\langle -\vec{\beta}, -\vec{\gamma} | H_C | -\vec{\beta}, -\vec{\gamma} \rangle = \langle \vec{\beta}, \vec{\gamma} | H_C | \vec{\beta}, \vec{\gamma} \rangle$.

Flipping signs of all angles for an instance, and/or shifting all β 's or all γ 's by multiples of π , we plot angles according to their deviation from 0. With the angle quantity visualizations having been clarified, and standardized so as to be comparable, using these mathematical transformations, the two sets of angle plots show a number of notable observations in regard to angle trends:

(1) The $\vec{\beta}, \vec{\gamma}$ angle schedules start from, and finally approach, particular values:

(a) $\vec{\beta}$ begins at around $\beta_1 \approx 0.6$ and approaches $\beta_p \approx 0$, converging closer to those values as p increases.

(b) $\vec{\gamma}$ begins at around $\gamma_1 \approx 0$ and approaches $\gamma_p \approx -0.35$ (equivalently ≈ 2.79), converging closer to those values as p increases.

This convergence is mostly only seen in the 16-qubit instances because of the available data going to much higher depth p .

(2) $\vec{\gamma}$ starting at $\gamma_1 \approx 0$ (and increasing in absolute value), with $\vec{\beta}$ approaching $\beta_p \approx 0$ (by decreasing in absolute value), resembles an adiabatic schedule, but we do not have an explanation of the close agreement of specific values of γ_p, β_1 across different schedules.

(3) Another quantity we can examine is the absolute range of the values in $\vec{\beta}, \vec{\gamma}$. These quantities correspond to the effective simulation time that was taken at each QAOA step. We can see that $\vec{\beta}$ spans a wider range of values than $\vec{\gamma}$.

(4) The slope of the QAOA schedules begin at low p depth being large, and then as p grows larger the slope becomes flatter, with each subsequent p taking a smaller time step.

(5) The overall shape of the learned QAOA angles show that for each increment of p , the QAOA schedule, while similar, does “drift” slightly. Still, the consistent trend means that we could perform regression on these learned QAOA

schedules (such as in Ref. [20]) in an attempt to extrapolate good QAOA schedules up to larger p .

The limitation with this approach is the convergence behavior mentioned above, which occurs as the expectation value approaches 1. However, for good extrapolation onto unseen larger problem instances the extrapolated QAOA angles would need to *not converge as quickly* as on the smaller instances. QAOA parameter extrapolation—beyond parameter transfer—would be a valuable future research question to probe.

(6) Finally, some of the trained QAOA angles have very different behavior that is not seen in the other problem instances—namely, instance number 5 in Fig. 1 shows highly fluctuating angles for the largest (converged) p that was found. This highly fluctuating instance could be related to the convergence to nearly an approximation ratio of 1.

Notably, these qualitative trend agreements of the QAOA angles between different optimization problem instances have been observed before in several other studies that have applied the QAOA to other types of optimization problems [20,50,51,84,85]—this suggests that good angles can be extrapolated using computationally simple polynomial fits based on a smaller number of trained p steps.

More generally, these optimized QAOA schedules closely follow recent studies that have drawn connections between highly optimized QAOA trajectories, and continuous-time quantum annealing schedules, which typically result in smooth QAOA angle trajectories [52,86–89] (this connection can be understood as QAOA being a type of discretized continuous-time adiabatic quantum annealing). These highly optimized QAOA angles, shown in Figs. 1 and 2, therefore provide more evidence for this fundamental connection between continuous-time adiabatic evolution and the discretized QAOA algorithm. This smooth-schedule property could be due to the underlying angle extrapolation technique used in the training; however, because the ground-truth of optimal QAOA angles are not known, there is no way to know whether these angles are optimal or not and therefore we consider the JuiQAOA angles to be good *heuristic* QAOA control parameters. Whether the QAOA schedule characteristics we observe are due to an artifact of angle extrapolation or not, their performance on large-scale problem instances shows that these types of QAOA schedules work very well, supporting similar findings from prior studies [52,86,87].

Figure 3 plots the mean noiseless approximation ratios for the set of trained QAOA angles shown in Figs. 1 and 2, evaluated on the problem instances on which the QAOA parameters were trained. Figure 3 shows the rate of energy improvement as a function of p (which quickly plateaus at large p for the 16-qubit instances), and moreover shows that the trained QAOA parameters give continual solution improvement as a function of p .

The 27-qubit instances of Figs. 2 and 3 (right) were trained only up to $p = 7$ because of the high classical computational cost required to train these instances up to very high rounds—the 16-qubit instances of Figs. 1 and 3 (left) were significantly easier to train because the size of the state vector simulation is significantly smaller.

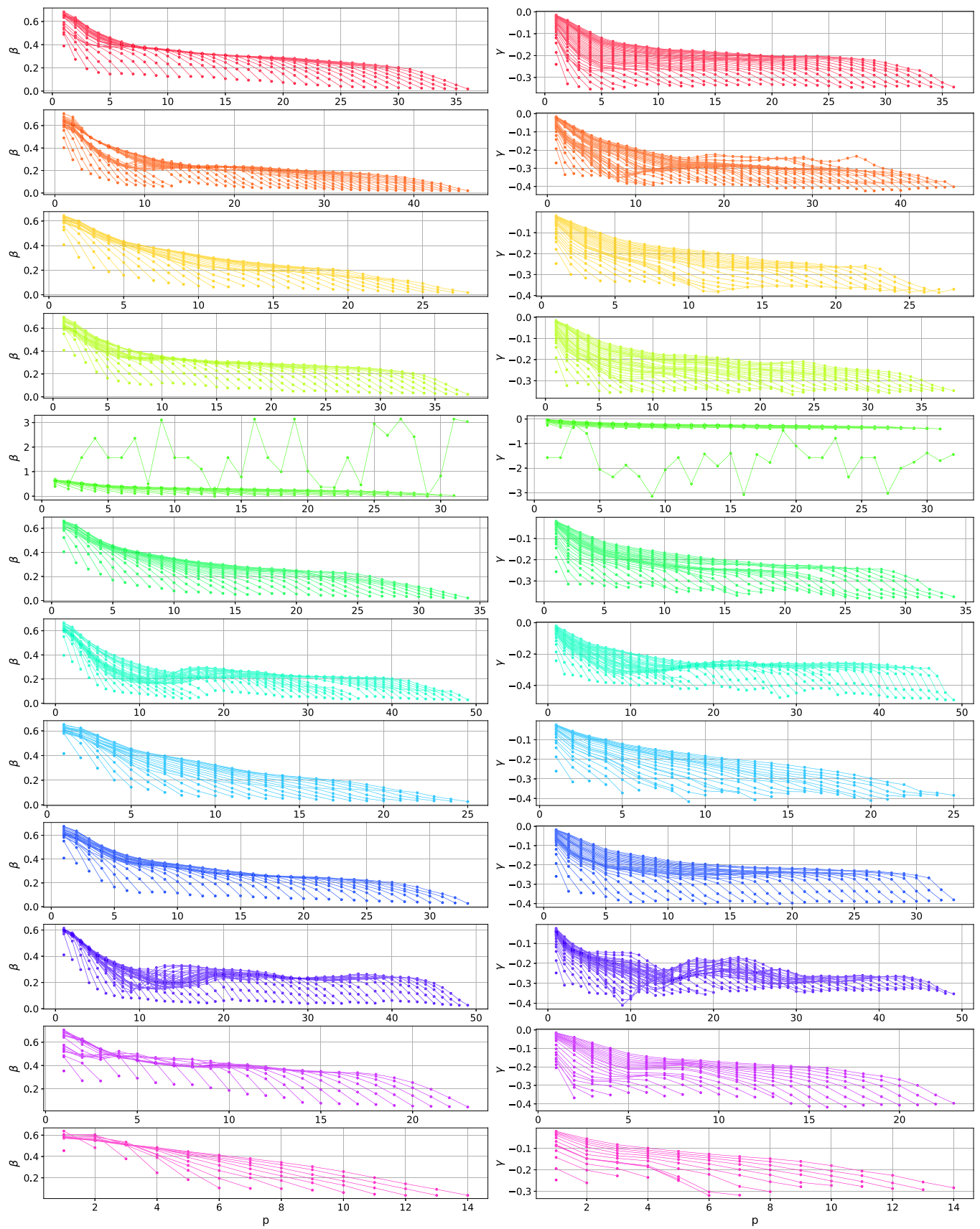


FIG. 1. Sixteen-qubit instance QAOA angles learned up to $p = 49$ or close to the ground-state energy. Each row corresponds to a different problem instance (first ten are random coefficients, and bottom two are uniform positive and negative coefficients).

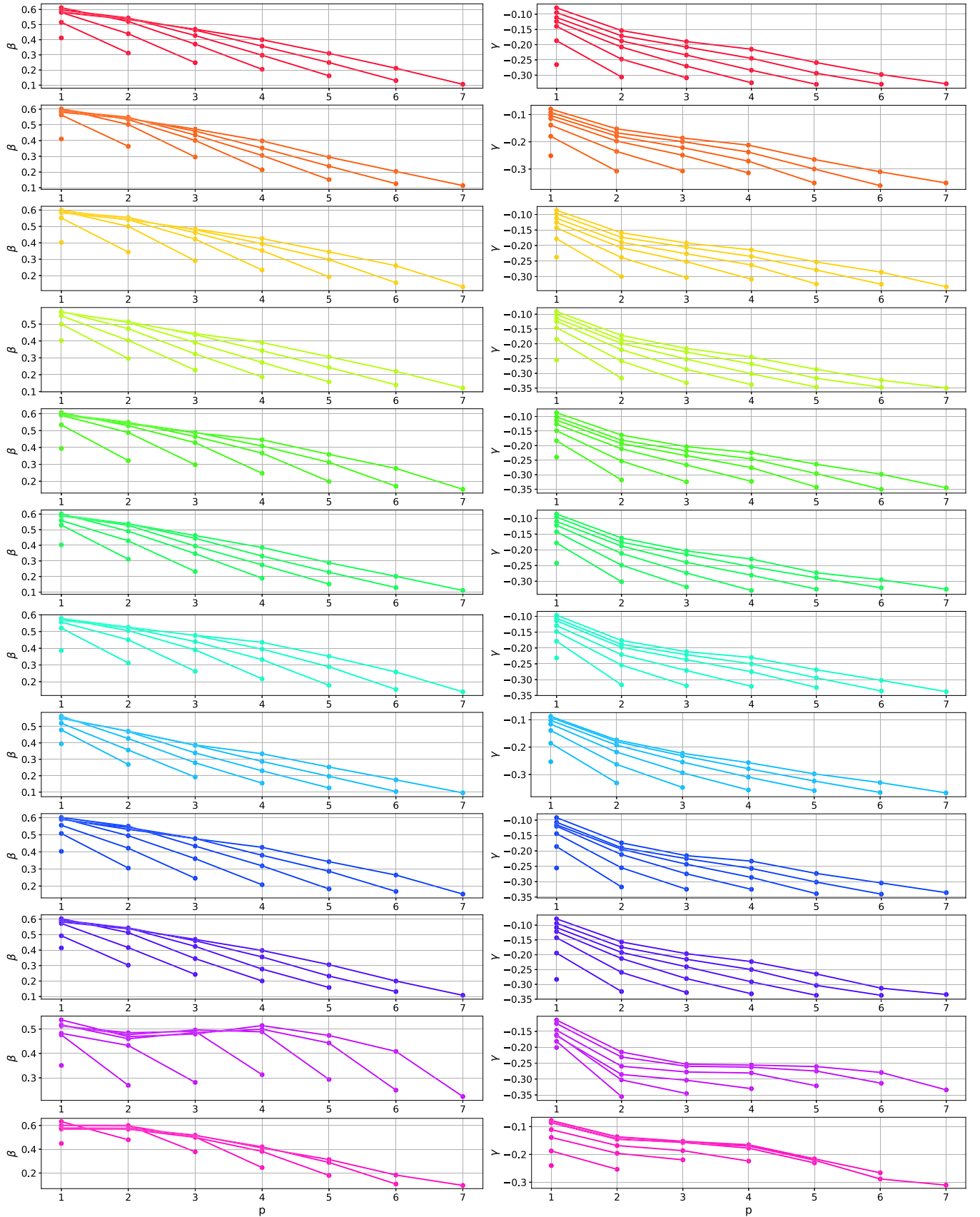


FIG. 2. Twenty-seven-qubit instance QAOA angles learned up to $p = 7$ for all instances. Each row corresponds to a different problem instance (first ten are random coefficients, and bottom two are uniform positive and negative coefficients).

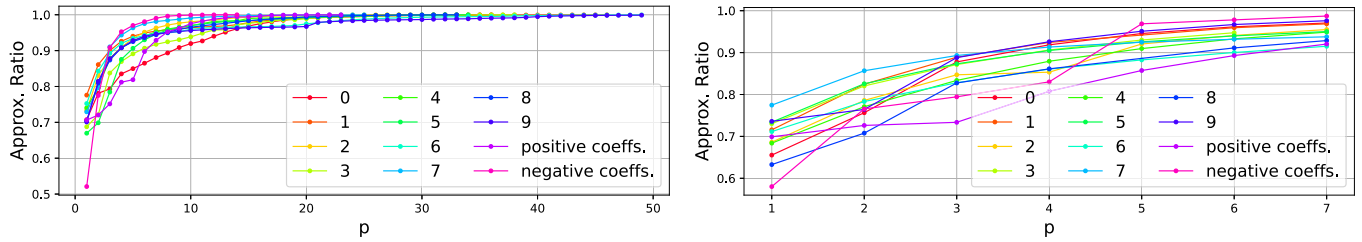


FIG. 3. Approximation ratio (y axis) vs p (x axis) for the 12 distinct 16-qubit Ising models (left) and the 12 distinct 27-qubit Ising models (right). The legends show the specific instance type—ten random-coefficient models and one model with entirely positive coefficients and one model with entirely negative coefficients. The 27-qubit instance training is cutoff at $p = 7$; note that for all instances the approximation ratio is above 0.90 at $p = 7$.

B. Validating small-scale parameter transfer with classical state vector simulations

Figure 4 shows, as a function of p , what the mean energy is when using a fixed set of QAOA angles learned from one of the (random-coefficient) 16-qubit instances on all of the other 16- and 27-qubit problem instances (which this set of fixed angles was specifically not trained on). These energy curves remarkably show that despite several blips of nonimproving energy as a function of p , there is a very consistent energy improvement up to very large p . Fundamentally, this shows that the limits of parameter transfer, at least for this size of problem instance (in terms of what size of p can still be used effectively), are the saturation of an approximation ratio of 1 on the training instance—if you are willing to accept temporary nonideal blips of nonconsistent improvement as a function of p . We argue that parameter transfer works if the performance (computed expectation value) is monotonically improving as a function of p ; this is the same requirement we impose on the initial QAOA training. However, if this condition is not met, parameter-transferred angles could still work well if the performance on average tends to improve at larger p relative to smaller p despite transitory decreases in performance. Having trained an ensemble of 24 sets of QAOA “schedules,” one approach could be to compute average quantities for these learned angles, which could represent a good schedule for an average problem. However, here we utilize all 24 sets of

angles to sample the underlying heavy-hex Ising models in order to obtain a spectrum of QAOA performance—where some outlier angles perform worse, but potentially others perform better. This is seen very clearly in Fig. 4, where the performance of the fixed angles differs significantly across the different instances.

Appendix B reports small-instance size parameter transfer numerical simulation results in Figs. 14 and 15, where the donor instances are entirely positive and entirely negative coefficient Ising models. The angles from these uniformly weighted models show a similar story where the transferred angles work well, but in some cases and at some p depths the approximation ratio improvement is not strictly monotonic. Moreover, the entirely positive-weighted coefficient Ising model training angles do not perform as well when transferred to other instances compared to the all negative coefficient model.

C. IBM quantum computer sampling using the transferred QAOA angles

Figures 5 and 6 plot the finite-shot estimates of energy expectation value as a function of the (learned) increasing p angles when executed on 127-, 133-, and 156-qubit IBM QPUs. There are only two differences between the 24 distinct connected lines in the scaling plots as a function of p : (1) time, meaning that there is possible time-dependent noise

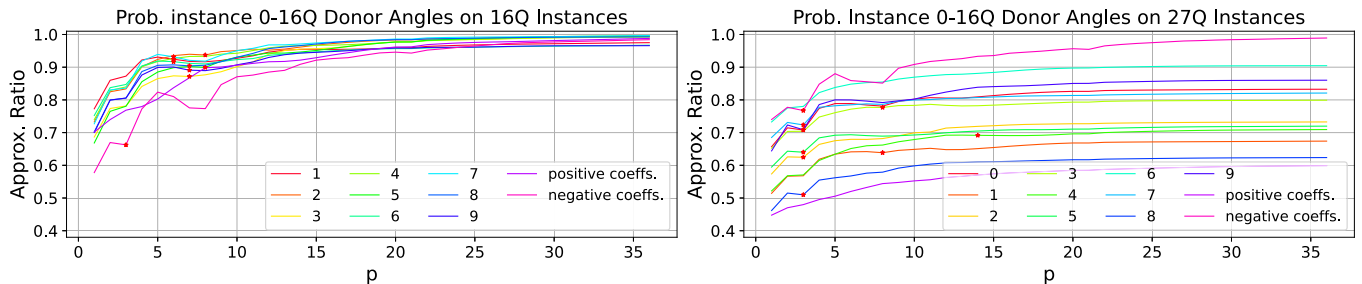


FIG. 4. Numerical simulation of parameter transfer of one of the 16-qubit problem instances (random coefficients) onto the other 11 16-qubit problem instances (left) and onto the 12 27-qubit problem instances (right). These simulations are all noiseless (no shot noise). y axis is the approximation ratio for the given target problem instances, and x axis is p QAOA steps. Each line corresponds to a different problem instance (which were not used in the training of this set of fixed angles), which is indexed in the legend numerically and by all the positive and negative coefficient Ising models. Red asterisk markers denote the p step where the expectation value decreases in quality (increases in energy) for that given problem instance, if that occurs. Eight out of the 11 instances in the left plot have at least one nonimproving p step, and 9 out of 12 of the instances in the right plot have at least one nonimproving p step. However, there is an overall trend for all problem instances shown in these plots that the higher p angles do give consistently improving energies, even if there are some blips of nonmonotonic improvements.

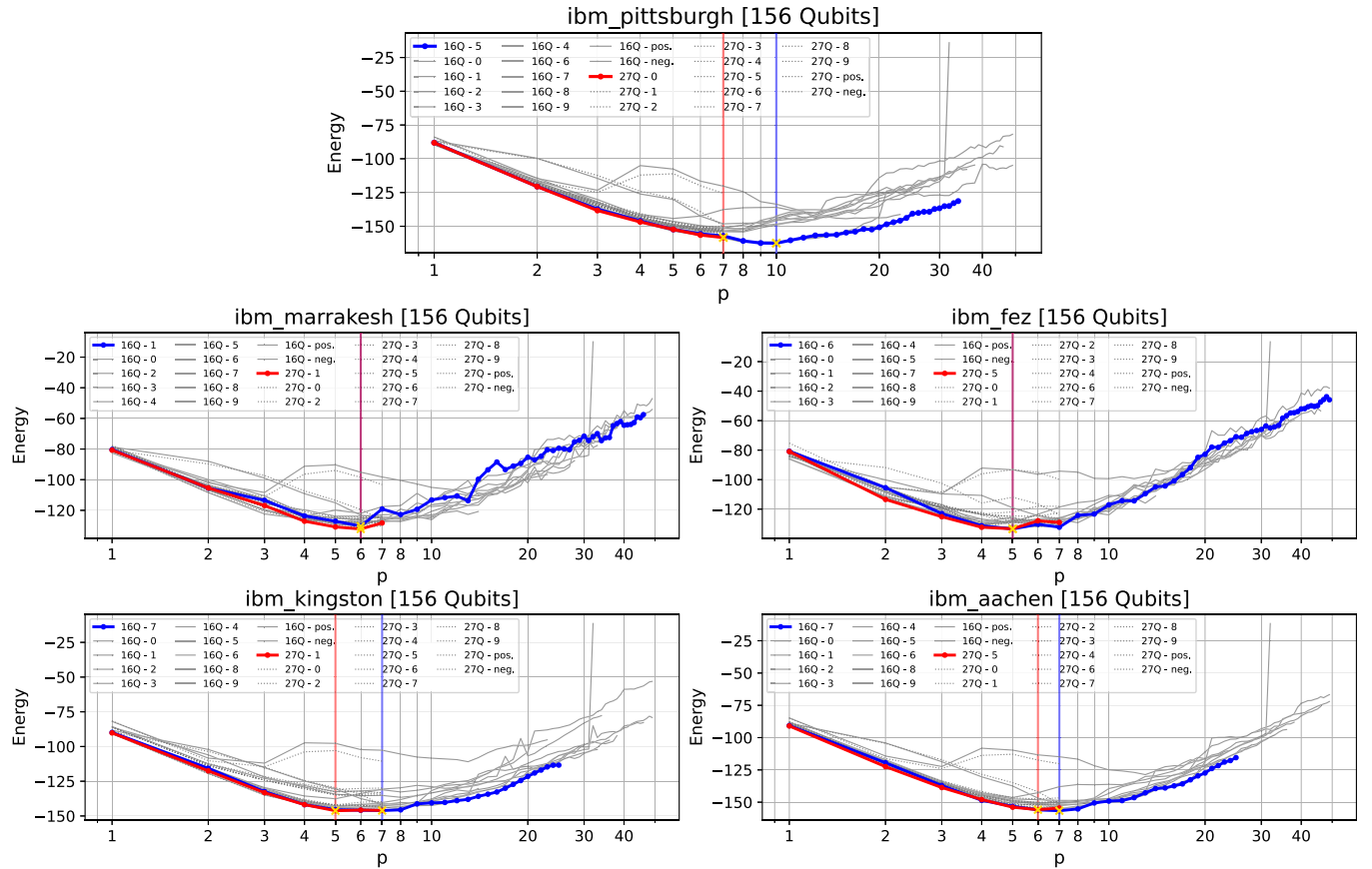


FIG. 5. IBM quantum processor QAOA performance results for the 156-qubit IBM quantum computers in terms of objective function (Ising model) energy on the y axis as a function of the QAOA depth p on the log scale x axis. Each processor is sampling a single 156-spin hardware-defined Ising model instance, whose global minimum energy is -246 . Each plotted point (defined by a p and a fixed set of transferred QAOA parameters) is showing the mean expectation value estimated using 10^5 samples, except *ibm_fez* where 10^6 samples are used. The classically trained angles are denoted in the legend by the qubit count of the problem followed by an index integer of the random instance (or all positive or negative coefficients), resulting in a total of 24 sets of QAOA angles. Within each subplot, the red and blue lines each show the two sets of transferred QAOA angles, for 27- and 16-qubit source instances, which resulted in the best (lowest) expectation value found on that particular QPU. Although these five subplots show sampling results from the same fixed Ising model, the y axes are not scaled to be the same for each device so as to highlight any small differences between the noisy computations. Note that each figure legend contains multiple lines that are either solid (16-qubit training instance) or dashed (27-qubit training instance) gray—these lines are not visually differentiable and are intended to show the collective sampling rates for the sets of fixed QAOA angles. The minimum (mean) energies, with continuous energy improvement as a function of p , are marked with $\times$ symbols and vertical lines at that p index.

drift on the quantum hardware (which does certainly occur on these QPUs [90,91]), and (2) performance differences of the transferred QAOA angles. The ensemble of learned angles are all unique (although, they are typically quite similar) because they were all trained on distinct random heavy-hex spin glass instances. The underlying combinatorial optimization problem instance that is being optimized is fixed, as is the QPU, as is the structure of the circuit being defined strictly on the heavy-hex graph (only single-qubit gate rotations are changed across the different compiled circuits from the different sets of angles). Therefore, the intention is for these plots to show the relative differences between the IBM QPUs across the set of 24 distinct sets of QAOA parameters (i.e., QAOA schedules) as the QAOA circuit depth increases, up to the largest parameter-transferred p depth. There is not a single source problem instance that resulted in trained QAOA angles that resulted in the best performance on NISQ hardware; instead, Figs. 5 and 6 show that the on-hardware performance is quite

similar across the different sets of angles (this is very likely due to the error rates, and likely biased errors, on the QPUs).

Figure 6 shows that there is a clear and consistent loss of signal where the mean expectation value converges to an energy of 0 for the 127-qubit processors when p is sufficiently large (around $p = 30$). Remarkably, however, both the 133- and 156-qubit processors (see Fig. 5) show a nonzero signal up to the deepest QAOA circuits tested, which were up to $p = 49$. This is an indication that the qubit coherence times are significantly better for the Heron processors compared to the 127-qubit Eagle processor devices. Although these extremely high-depth QAOA circuits at $p = 49$ do not show a continual improvement of mean expectation value (because of noise), these are the highest depth QAOA digital circuits executed on noisy quantum processor to date—and it is notable that there is a nonzero signal generated by some, but not all, of the IBM quantum processors. It is surprising that despite the peak performance of the 156-qubit IBM Heron r2 and r3 devices

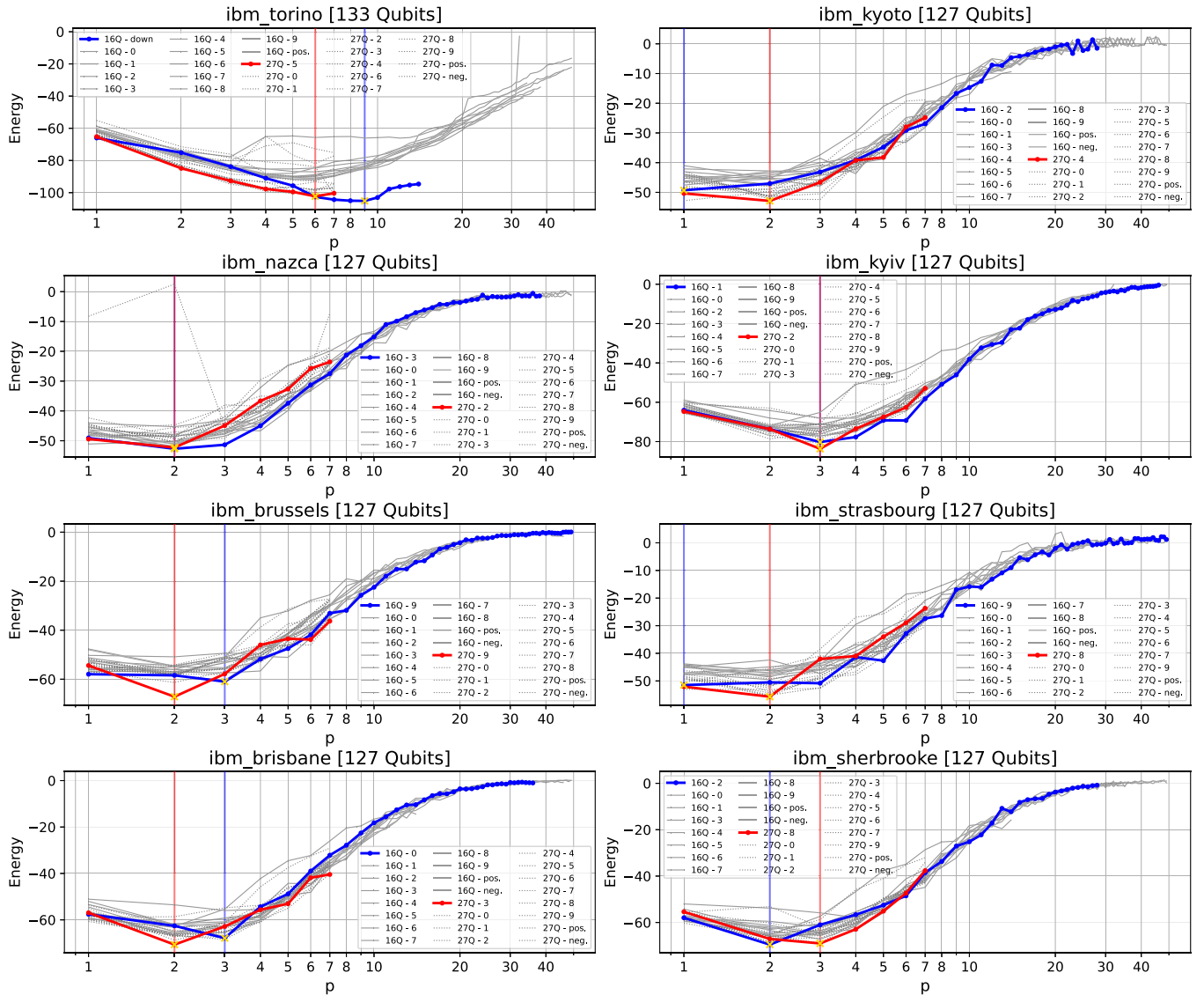


FIG. 6. IBM quantum processor QAOA performance results in terms of objective function (Ising model) energy on the y axis as a function of the QAOA depth p on the log scale x axis, where the parameters used for each p are the same angles that were learned on the smaller problem instances classically. This is a continuation of Fig. 5 using the same format, but for different IBM devices. The classically trained angles are denoted by the qubit count of the problem followed by an index integer of the random instance (or all positive or negative coefficients), resulting in a total of 24 sets of QAOA angles. The optimization problem is a minimization task, meaning that lower energies are better solutions. Each point (defined by a p and a fixed set of transferred QAOA parameters) is showing the mean energy sampled from 10^6 measurements, with the exception of *ibm_torino* where only 10^5 were measured. The set of angles that resulted in the lowest (best) mean energy for both the 16-qubit and the 27-qubit instances are shown as blue and red lines, respectively. The minimum (mean) energies, with continuous energy improvement as a function of p , are marked with $\times$ symbols and vertical lines at that p index. The optimization problem being solved by each QPU is identical for each QPU size (since the hardware two-qubit entangling gate connectivity structure is exactly the same for each QPU size); the instance is the same for all of the 127-qubit devices, and there is a single 133-qubit instance for *ibm_torino*.

being at $p \approx 10$, there is a nonrandom Hamiltonian expectation found all the way up to $p = 49$. For quantum circuit size context, the largest one of these QAOA circuits (both with respect to qubit count and circuit depth) uses 156 qubits, 17 248 CZ two-qubit gates, 40 924 SX single-qubit gates, 34 513 RZ single-qubit gates, and 156-qubit measurement operations at the end of the circuit. This qualitative result is made especially clear by the *ibm_pittsburgh* data, where the largest QAOA circuit expectation value at $p = 49$ is a lower energy (better) than, or approximately equal to, the $p = 1$ results.

These QAOA simulations on NISQ computers required a large amount of QPU time, which is summarized in Table II. Note that these compute times depend heavily on the total number of samples (e.g., circuits executed). These total compute times give a clear perspective on how efficient the different QPUs are (taking into account the lower shot counts on several of the devices)—for example, that *ibm_fez* and *ibm_marrakesh* are faster than all of the 127-qubit devices (i.e., consumes less QPU time per sample).

TABLE II. IBM NISQ computer QAOA hardware-compatible Ising model sampling summary. The *min sample energy* is the absolute lowest energy found on that QPU, across all QAOA parameters. The corresponding ground-state energies are -194 (for 127 spins), -196 (for 133 spins), and -246 (for 156 spins). The *total QPU time* is the absolute sum of QPU time (server side, not including local compilation and circuit construction) used across all QAOA circuit executions on that QPU (data not available for *ibm_kyoto* and *ibm_nazca*). The *measurement count* is the number of shots sampled for each p , for each of the 24 QAOA angle sets. Note that this constitutes different numbers of backend jobs—for all 127-qubit processors and *ibm_fez*, this is 10 distinct backend jobs, for *ibm_torino* this is 20 jobs, and for the remaining three 156-qubit devices, this was a single backend job. Finally, the last column denotes the *best QAOA volume* that was found, in terms of the mean expectation value of the best performing QAOA angle with continuous improving performance at higher-depth circuits, in the form of $p \times N$ where p is the QAOA depth and N is the number of discrete variables.

QPU size	QPU name	Energy		QPU time, <i>total</i> (s)	Measurement count, <i>per</i> parameter comb.	QAOA volume, <i>best</i> ($p \times N$)
		Ising GS.	Min sample			
127 qubits	<i>ibm_sherbrooke</i>	-194	-160	192 740	10^6	$3 \times 127 = 381$
	<i>ibm_kyiv</i>	-194	-168	415 753	10^6	$3 \times 127 = 381$
	<i>ibm_brisbane</i>	-194	-156	181 311	10^6	$3 \times 127 = 381$
	<i>ibm_kyoto</i>	-194	-144	n/a	10^6	$2 \times 127 = 254$
	<i>ibm_nazca</i>	-194	-142	n/a	10^6	$2 \times 127 = 254$
	<i>ibm_brussels</i>	-194	-154	176 812	10^6	$3 \times 127 = 381$
	<i>ibm_strasbourg</i>	-194	-146	176 526	10^6	$2 \times 127 = 254$
133 qubits	<i>ibm_torino</i>	-196	-178	41 331	10^5	$9 \times 133 = 1197$
156 qubits	<i>ibm_fez</i>	-246	-226	156 322	10^6	$5 \times 156 = 780$
	<i>ibm_marrakesh</i>	-246	-218	16 723	10^5	$6 \times 156 = 936$
	<i>ibm_kingston</i>	-246	-228	14 171	10^5	$7 \times 156 = 1092$
	<i>ibm_aachen</i>	-246	-238	15 825	10^5	$7 \times 156 = 1092$
	<i>ibm_pittsburgh</i>	-246	-232	14 459	10^5	$10 \times 156 = 1560$

Notably, none of the NISQ computer QAOA distributions shown in Fig. 6 sampled a globally optimal energy spin configuration, despite the usage of a wide range of trained parameter transfer angles, in addition to high measurement counts. Table II reports the lowest energies sampled on each QPU.

In order to show the component level error rates on these IBM NISQ processors used in Figs. 5 and 6, Fig. 7 shows aggregate single- and two-qubit gate error rates, as well as the qubit coherence time metrics of $T1$ and $T2$ times across most of the IBM NISQ computers. $T1$ is the relaxation time and $T2$ is the dephasing time [92]—these metrics give estimates

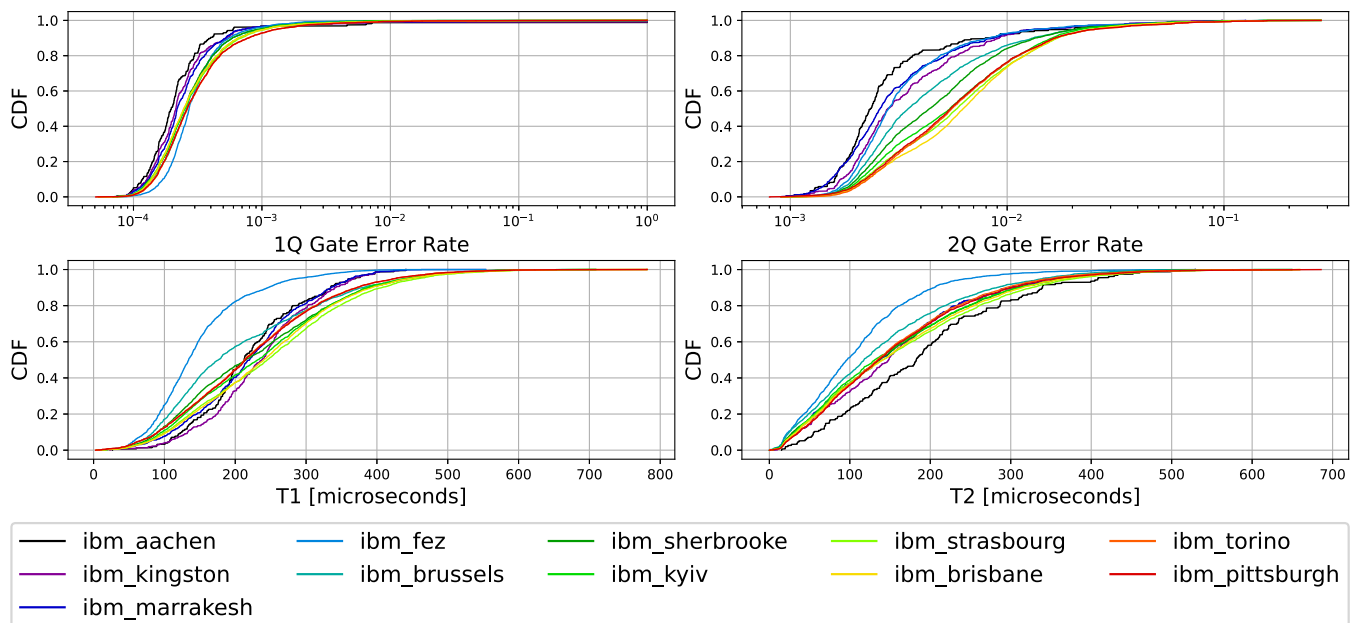


FIG. 7. IBM quantum processor hardware error rates, aggregated from the vendor-provided calibration data for all QAOA circuits that were executed and whose results are shown in Figs. 5 and 6. Top row shows the single- and two-qubit gate randomized benchmarking error rates (log scale x axis), and bottom row shows $T1$ and $T2$ coherence times (x axis is in microseconds). Each plot presents the cumulative distribution function (y axis) of the combined hardware measured error rates for each device.

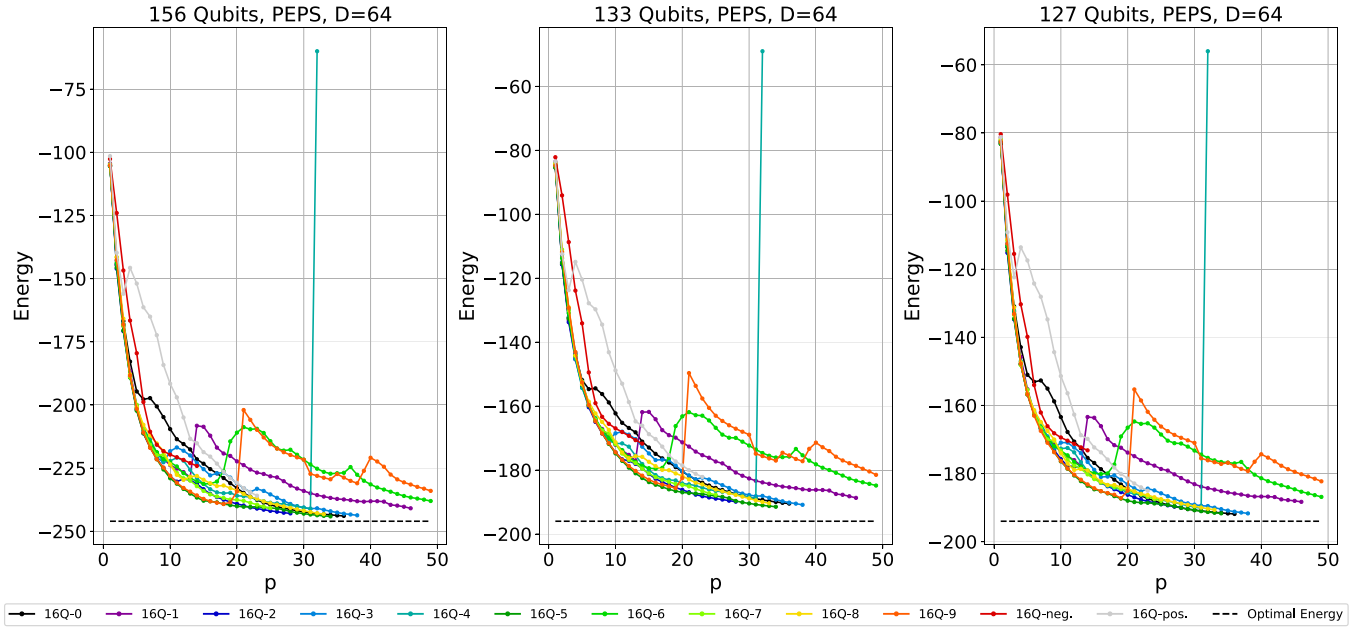


FIG. 8. Validating the QAOA angle quality, trained on the set of 12 16-qubit instances, applied to the 156-qubit (left), 133-qubit (middle), and 127-qubit (right) instances, using $D = 64$ PEPS simulations with belief propagation for approximate tensor network contraction. The different fixed QAOA angles are denoted by the 12 different lines shown on the plots, for example, 16Q-pos. denotes the set of QAOA angles learned on a 16-qubit heavy-hex Ising model that has entirely positive coefficients. The true ground-state energy of each Ising model is denoted by a dashed black line.

for how long the qubits remain coherent (the larger these times are, the better the qubits are). These error rates are reported by the device backends, and here we are showing the cumulative distribution function of the distribution of all error rates collected from every job that was submitted to the device (and that successfully executed) whose results are shown in Fig. 6—this means that the reported distribution technically contains duplicates because the IBM devices are only calibrated once every some set number of hours (dependent on the device), and these metrics are reported by this device calibration on the backend. The goal of showing this distribution in this way is because it shows the *aggregate effective error rates* encountered by the QAOA circuits seen in Fig. 6, and moreover show differences in qubit coherence times on the different hardware platforms. These metrics are measured using randomized benchmarking [93,94] techniques. Note that *ibm_nazca* and *ibm_kyoto* are not included in this dataset because the hardware level error rates could not be extracted from the Qiskit API postcircuit execution. The gate error rates shown in Fig. 7 are based on recording the error rates of the gateset to which the circuits are compiled to. The error rates of the single-qubit gates are recorded based on the gates SX and X, and the error rates of the two-qubit gates are recorded based on the gates ECR and CZ. Note that the single-qubit gate RZ is a *virtual* gate that is implemented as an abstract circuit compilation operation [95], which means that the error rate for this gate is always 0. When retrieving two-qubit gate error rates from the IBM backends, there are many instances of reported error rates of 1.0. It seems that these reported error rates are indicating either an uncalibrated two-qubit gate or a gate that has been deactivated at the software level. Because these error rates are an erroneous backend related artifact, we have discarded all 1.0 error rates in the data shown in Fig. 7.

Figure 7 shows that the $T1$ and $T2$ times of the newer generations of IBM processors, such as *ibm_marrakesh*, are smaller than the previous generation QPUs. However, this does not mean that the newer devices actually have worse performance—the reason is because the single- and two-qubit gate times were also reduced, along with the coherence times, on these Heron processors, and the relevant measure therefore becomes circuit depth that can be executed within the coherence times.

One way of quantifying the size of QAOA circuits that have been executed on noisy quantum computers such that there is continual improvement at each step of p is a concept called “QAOA volume” [10,11] that defines the largest QAOA circuit from N variables and depth p that results in monotonic improvement of solutions up to that depth p (analogous to the quantum volume NISQ computer benchmark [96–99]). The QAOA volumes reported in the hardware simulations shown in Figs. 5 and 6 are summarized in Table II. These are some of the largest QAOA volumes reported on noisy quantum computers to date, going up to a continual improvement of $p = 10$ on a 156-qubit processor. QAOA volume here shows a benchmark of current IBM NISQ computer capability.

D. Validating large-scale QAOA parameter transfer with PEPS simulations

Figure 8 plots the Hamiltonian expectation value as a function of p , for the complete set of 12 trained sets of QAOA angles, using PEPS with a bond dimension of $D = 64$, for the 127-, 133-, and 156-qubit Ising model instances. The goal here is to validate whether the parameter transfer continues to work at larger problem sizes. Indeed, we see that for some of the QAOA angles, parameter transfer works remarkably

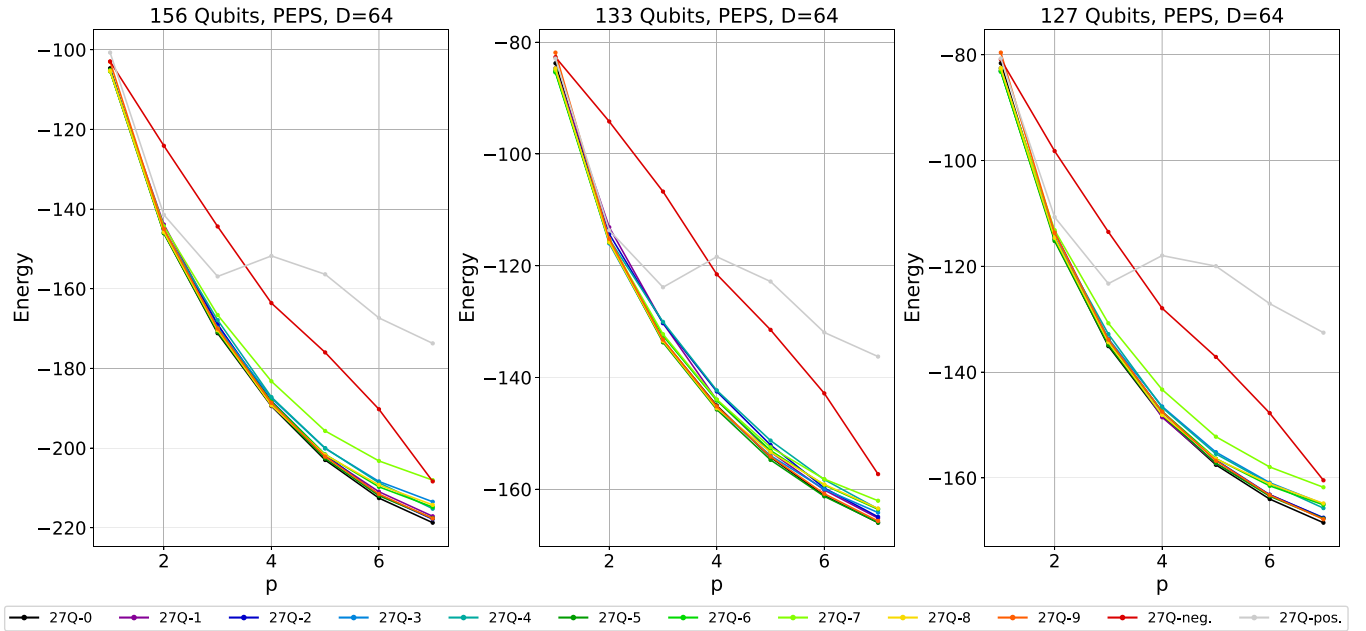


FIG. 9. Validating the QAOA angle quality, trained on the set of 12 distinct 27-qubit Ising model instances, applied to the 156-qubit (left), 133-qubit (middle), and 127-qubit (right) instances, using $D = 64$ PEPS simulations with belief propagation for approximate tensor network contraction. Ground-state energy not marked because these expectation values ≤ 7 do not reach near the ground state, as opposed to Fig. 8 where the mean expectation value does approach the true ground-state energy.

well—allowing convergence to nearly the true ground state. However, similarly to the smaller-scale state vector simulations of Figs. 4, 14, and 15, we see that for some sets of parameters, there exist nonmonotonic fluctuations, dips, and in one case there is a single set of QAOA angles that results in a significantly worse performance (note that this angle is shown explicitly as the highly fluctuating schedule shown in the fifth row from the top of Fig. 1). This result shows a single case where nonsmooth QAOA angles can perform well on the small test case problem instance, but then fail to perform well when those parameter are transferred up to larger problem sizes—we hypothesize that this is likely a rare artifact associated with having converged to close the ground-state energy, although additional investigation is required on this point.

In summary, Fig. 8 shows that QAOA parameter transfer can work well across instances that have a parameter-transfer Ising model size difference (between the trained instance and the tested instance) of up to 140 discrete variables, and in this case allows us to get near to the true ground state, assuming a noiseless quantum computation (approximated by PEPS with a bond dimension of $D = 64$).

Figure 9 shows similar trends of monotonic expectation value improvement as a function of p from the 27-qubit instance trained angles, as we would want from good QAOA evolution—but with one exception of the entirely positive coefficient Ising model having a nonmonotonic dip in performance. Comparing Figs. 8 and 9, there is little difference between the angles that were trained on either the slightly smaller (16-qubit) or larger (27-qubit) instances.

Interestingly, one of the primary conclusions from all of these QAOA parameter transfer calculations is that overall, coefficient choice does not significantly impact the transfer of parameters (the only clear exception to this is that

parameters from the positive coefficient case do not monotonically perform well on the other instances). Even very extreme coefficient choices, and random-coefficient choices, at sufficiently large p tend to work well. This suggests that these parameters could work well because the Hamiltonian connectivity structure was the same for all instances; however, we do not evaluate parameter transfer across different connectivity graphs.

Figure 10 plots the relative expectation value differences from PEPS simulations at different bond dimensions run on the largest Ising model we consider in this study (156 spins). This shows that the tensor network simulations clearly converge at small p (typically $p \leq 7$) and typically fail to converge as strongly at larger p , with slower convergence occurring in the intermediate values of p , typically $p = 10$ to 20. Moreover, the trend of PEPS convergence is quite distinct for each set of fixed QAOA angles. This empirical convergence evidence reflects what we would expect from highly optimized QAOA evolution [100], similarly to quantum annealing, where in the middle of the evolution the system has high entanglement entropy, thus making it harder for the PEPS simulation to converge.

Figure 18 in Appendix E shows the convergence properties of the remaining set of trained QAOA angles, run on the single 156-qubit instance. Interestingly, this lack of full convergence at intermediate p values (typically, this is approximately between $p = 10$ and $p = 30$) suggests that those specific QAOA angles result in circuits that are harder to approximate classically using tensor network methods. This convergence trend manifests across most of the angles, with the one exception of the all negative-coefficient angle set shown in Fig. 18. However, the larger p simulations, which use slightly different QAOA angles, are then closer to convergence. This suggests

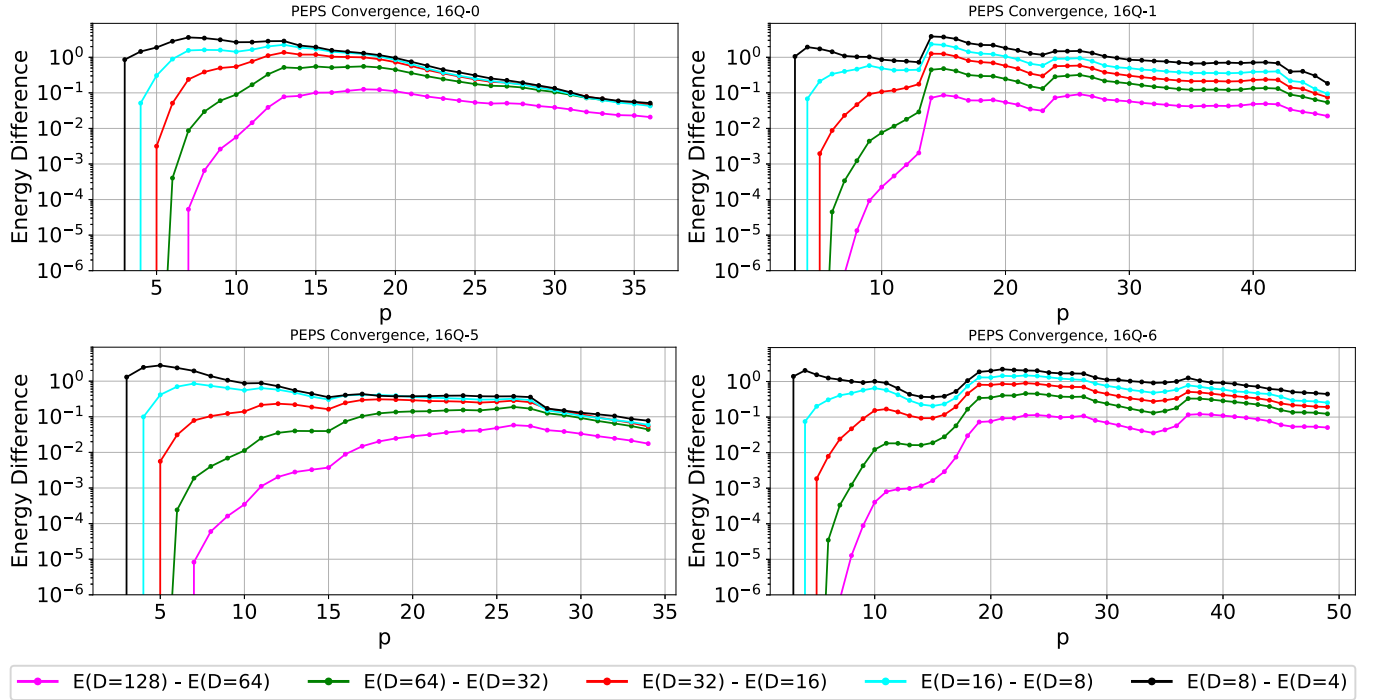


FIG. 10. PEPS simulation convergence analysis using four of the sets of parameter-transferred QAOA angles (specifically from random-coefficient Ising models) onto the 156-qubit hardware-compatible Ising model. Tensor network contraction is performed using belief propagation. The log-scale y axis plots the relative difference between energy estimates obtained with subsequent PEPS bond dimensions. Overlapping lines that are higher on the y axis indicate lack of convergence as the bond dimension increases, which occurs in particular for larger values of p , likely where higher entanglement occurs in the circuit (although, an entanglement measure is not quantified in these simulations). On the other hand, lines that become substantially smaller on the y axis as the bond dimension increases, for example, in cases of small p where the lines drop off the log-scale axis plots entirely, show clear numerical convergence of the Hamiltonian expectation value.

that there do exist particular QAOA parameter combinations for these circuits that are hard to classically simulate; however, unexpectedly the optimized angles, which typically approach the true ground-state energy, are easier to approximate classically. This could be related to the properties of the ground states themselves that are approached in the high- p regime, which can be expressed as a linear combination of many product states (i.e., due to high degeneracy of the Ising models).

Convergence of the PEPS simulations would resemble a clear progression downward on the y axis of logarithm expectation value difference in Fig. 10. Eventually, at very large bond dimension, convergence results in the difference between bond dimensions (which are increasing as powers of 2) that have zero difference between each other. This effect is really only seen very strongly at low- p depth in Fig. 10, and in some cases we see a strong lack of convergence at sufficiently high p , indicating those expectation values have higher uncertainty, even at $D = 128$.

Finally, a natural question in regard to quantum computation is the effect of sampling—for example, Fig. 8 showed only the mean expectation value from the PEPS simulation, namely, because all quantum computation requires measuring the state of the qubits at the end of the computation. Figure 11 shows sampling from the PEPS simulation at $D = 64$ using 1000 samples (for each QAOA angle, at each p), where we quantify the sampled ground-state energy proportion. Interestingly, this shows that some of the parameter transfer angles

work very well for ground-state sampling, typically well exceeding a proportion of 0.1, while others do not perform well. This suggests that in general for parameter transfer, using an ensemble of distinct QAOA angles is a good technique. However, the 133-qubit instance ground-state energy is very rarely sampled. The gap between the best-performing angle and the ground-state energy of the 133-qubit in Fig. 8 is larger than the gap for the other two instances, and therefore the low ground-state sampling rate in Fig. 11 is consistent with the noiseless expectation values. This suggests that the low GS sampling rate is an instance-dependent result, and the performance of these fixed angles likely has high variability across different random-coefficient problem instances. Figure 11 reports only the QAOA angles originating from the training on 16-qubit instances since the 27-qubit instances were only trained up to $p = 7$, and in that QAOA depth regime we verified with PEPS simulations that GS energy sampling rate (using 1000 samples per p) is zero.

E. Validating QAOA PEPS simulations with MPS simulations

The MPS simulations are performed with $\chi = 2^k$, for $k = 7, \dots, 11$ and shown in Fig. 12 for the 156-qubit QAOA circuits. Panels (a) and (b) display the error in the energy ΔE as a function of p for several values of MPS bond dimension χ . Here, the error ΔE quantifies the difference between our most accurate PEPS energy approximation and the MPS

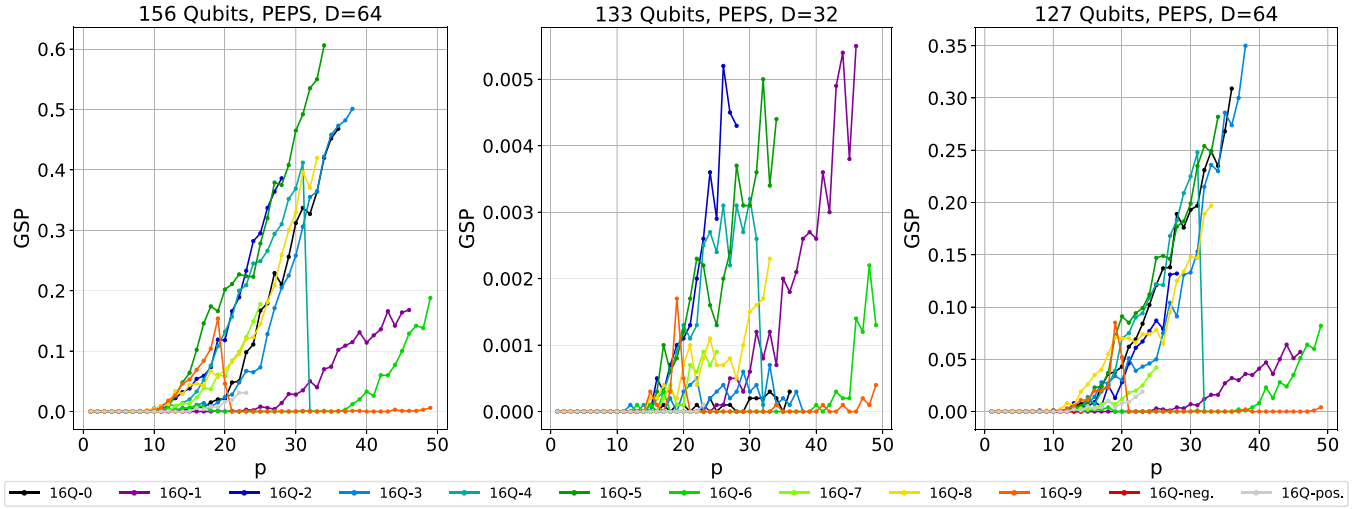


FIG. 11. Ground-state energy QAOA sampling rate (probability of sampling the ground state) from $D = 64$ and $D = 32$ PEPS simulations, across the ensemble of 16-qubit trained angles (notated by integer indexing shown in the legend), on the 156-qubit Ising model (left), 133-qubit Ising model (middle), and the 127-qubit Ising model (right). Interestingly, the 133-qubit instance ground state is rarely sampled, in comparison to the other two models. The ground-state sampling proportion is estimated using 1000 samples, per parameter (10 000 in the case of the 133-qubit case, to reduce the effect of shot noise due to the low ground-state energy sampling rate), and each subplot y axis is different for each Ising model.

approximation computed with a given χ . It is defined as

$$\Delta E = \left| \frac{E_{D=128}^{\text{PEPS}} - E_{\chi}^{\text{MPS}}}{E_{D=128}^{\text{PEPS}}} \right|, \quad (4)$$

where $E_{\chi}^{\text{MPS}} = \langle \psi_{\chi}^{\text{MPS}} | H_C | \psi_{\chi}^{\text{MPS}} \rangle$ and $|\psi_{\chi}^{\text{MPS}}\rangle$ is a bond-dimension- χ MPS approximation to $|\tilde{\beta}, \tilde{\gamma}\rangle$ in Eq. (1). Similarly, $E_D^{\text{PEPS}} = \langle \psi_D^{\text{PEPS}} | H_C | \psi_D^{\text{PEPS}} \rangle$, where $|\psi_D^{\text{PEPS}}\rangle$ is PEPS approximation to $|\tilde{\beta}, \tilde{\gamma}\rangle$ computed with a bond dimension D .

Panels (a) and (b) show the results obtained with angles from 16-qubit random-coefficient Ising model index 0 and 6. MPS simulations performed with $p \leq 2$ are exact with $\chi = 2048$ and indeed $\Delta E = 0$ for those values of p . For

larger p , the simulations are only approximate within the MPS framework and bond dimensions studied. Nevertheless, the errors ΔE are small and $\chi = 2048$ simulations drop below $\Delta E = 0.005$ for all values of p in the instances 16Q-0 and 16Q-6. The plots show systematic and regular decrease in error as a function of χ for all values of p . This is a strong indication of high accuracy and dependability of our largest- D PEPS simulations. MPS simulations closely recover the best PEPS results for some values of p but struggle in the hardest region of intermediate p . Overall, MPS results presented here demonstrate that (1) large bond dimension PEPS simulations are very accurate and reliable and (2) MPS, despite having to

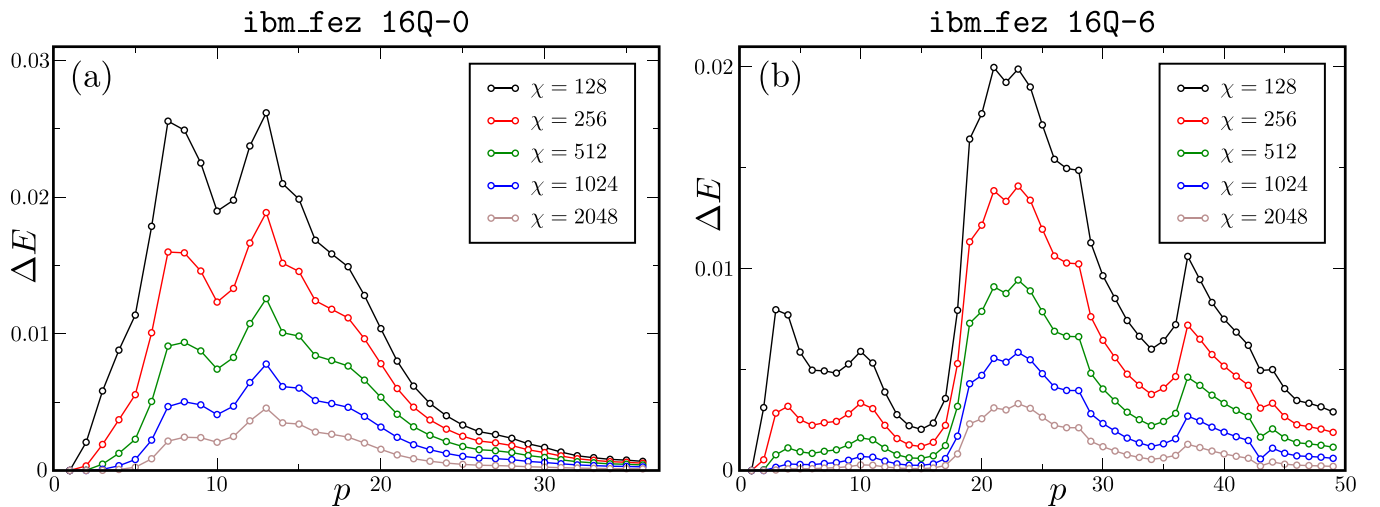


FIG. 12. Discrepancy between the most accurate PEPS simulation at $D = 128$ and multiple MPS simulations, for the largest QAOA circuit considered in this study of 156 qubits (defined on the `ibm_fez` hardware graph). The error in the energy ΔE , defined in Eq. (4), as a function of p is shown for several values of MPS bond dimension χ . The error ΔE decreases rapidly with increasing χ and drops below 0.005 for hardest to simulate values of p . The results confirm the reliability of PEPS simulation and establish $D = 128$ PEPS as our most accurate approximation to $|\tilde{\beta}, \tilde{\gamma}\rangle$ across all values of p .

handle highly nonlocal interactions, provides good accuracy across a broad range of p .

IV. DISCUSSION AND CONCLUSION

With the overall goal of developing nonvariational learning QAOA with the use of fixed parameters that work well on a general set of problem types, using parameter transfer by numerically learning on smaller problem instances is clearly a valuable approach to making QAOA not require significant resources for variational training. We have demonstrated this on a certain class of IBM NISQ computer hardware-compatible Ising models, using a combination of real NISQ computer hardware experiments and tensor network simulations. Notably, in the investigation of QAOA parameter transfer, we have experimentally demonstrated high-depth QAOA volumes of up to 1092 and 1197. Moreover, the ensemble of QAOA angles exhibits a general trend of improving expectation value when applied as transferred parameters, evaluated on up to 156-qubit instances.

However, parameter transfer does have limitations. Concretely, the following limitations exist for QAOA parameter transfer, primarily for the task of training on smaller problem instances and applying those parameters to larger problems. The most direct limitation is that once the QAOA training saturates the maximum approximation ratio of 1, meaning no further improvement is possible, then the training cannot supply good angles past that saturation point (for larger p) to use on any larger problem instances. This saturation is an issue specifically when we generally expect higher QAOA depth simulations to be required, to reach a certain threshold of accuracy (such as a ground-state sampling rate), as problem size increases. This limit necessitates the development of other techniques to be able to efficiently compute good QAOA angles, such as extrapolating to higher QAOA depths based on the already learned angles.

This previous limitation could be mitigated by finding fixed angle rules, or extrapolating QAOA angle trends, that heuristically work well, based on the QAOA training on smaller problem instances. However, such extrapolation would need to be carefully crafted for higher-depth QAOA circuits. Further study of using QAOA angle extrapolation based on QAOA parameter-transferred angles is warranted and could result in very scalable high-depth variational-learning free QAOA. Here, variational-free refers to the property that the fixed angles work reasonably well across different problem instances, even at high QAOA depth and on larger Ising model sizes—meaning, one does not need to reoptimize the QAOA parameters for every problem instance.

The transferred QAOA parameters do not always work well, in the sense they do not necessarily result in continual solution quality improvement as a function of p . Despite this, in the QAOA parameters we evaluated numerically there is a *general* solution quality improvement when using all transferred angles, despite local blips of noncontinual improvement. Additionally, QAOA parameter transfer does not always work well because of the variability that can result from training on specific problem instances—we observe a relatively large distribution of QAOA angle performances across the different trained instances. This then suggests that

parameter transfer can be applied in two distinct ways: either computing average angles from an ensemble of trained instances or using an ensemble of trained angles.

ACKNOWLEDGMENTS

We thank Stefan Woerner for constructive discussions. This work was supported by the U.S. Department of Energy through the Los Alamos National Laboratory. Los Alamos National Laboratory is operated by Triad National Security, LLC, for the National Nuclear Security Administration of U.S. Department of Energy (Contract No. 89233218CNA000001). Research presented in this article was supported by the NNSA's Advanced Simulation and Computing Beyond Moore's Law Program at Los Alamos National Laboratory. This research used resources provided by the Darwin test bed at Los Alamos National Laboratory (LANL) that is funded by the Computational Systems and Software Environments subprogram of LANL's Advanced Simulation and Computing program (NNSA/DOE). This research used resources provided by the Los Alamos National Laboratory Institutional Computing Program, which is supported by the U.S. Department of Energy National Nuclear Security Administration under Contract No. 89233218CNA000001. See LANL Report No. LA-UR-25-28746. We acknowledge the support of the National Science Center (NCN), Poland, under Projects No. 2020/38/E/ST3/00150 (M.M.R.) and No. 2022/47/D/ST2/03393 (P.C.). The research was also supported by a grant from the Priority Research Area DigiWorld under the Strategic Programme Excellence Initiative at Jagiellonian University (M.M.R. and P.C.). We acknowledge the use of IBM Quantum services for this work.

The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Quantum team.

DATA AVAILABILITY

The data that support the findings of this article are openly available [80,101].

APPENDIX A: DETAILED QAOA CIRCUIT DESCRIPTION

Figure 13 gives an explicit description of the QAOA circuit formulation that is used in this study to sample this specific class of current IBM-processor-defined two-qubit gate interaction graph. This QAOA circuit description can be extended to arbitrary p by repeating this same $p = 1$ structure. All QAOA simulations used the standard X mixer. All qubits are measured at the end of the simulation in the computational Z basis.

The QAOA circuits were all compiled with optimization using Qiskit [63,64] prior to submission to the QPUs—note, however, that in terms of two-qubit gate complexity, these circuits are already very heavily optimized, by hand compilation. Therefore, the primary function this served was to adapt the circuits to the correct gateset.

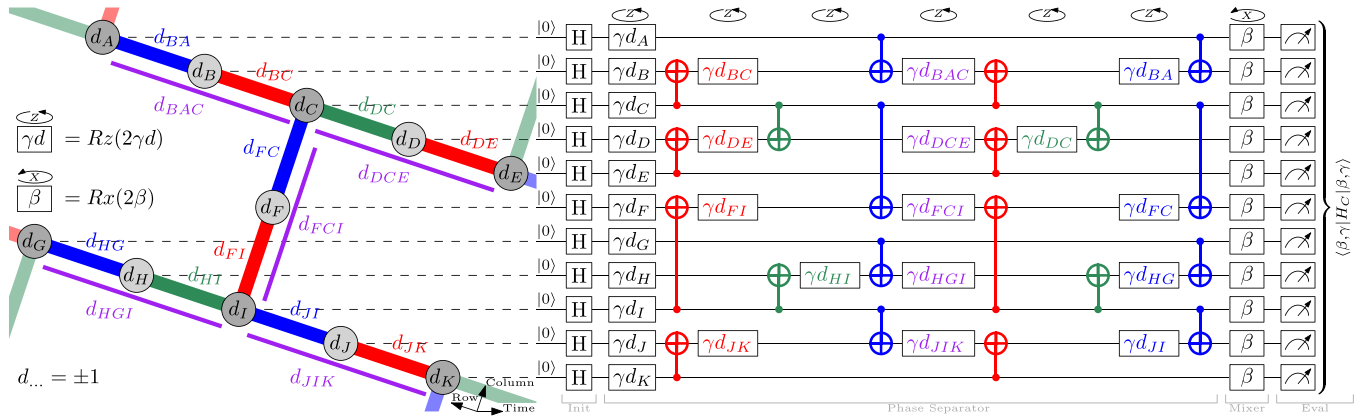


FIG. 13. Heavy-hex-defined Ising model with geometrically local cubic terms (left), corresponding $p = 1$ QAOA circuit description (right). Geometrically local cubic terms are denoted by the short purple lines that are adjacent to lines of three qubits. Red, blue, and green edge coloring denotes an example unique 3-edge-coloring of the lattice. Light and dark gray node coloring denotes a heavy-hex lattice bipartition. The coefficient weights of the local fields and of the quadratic terms are denoted as d_x . This QAOA circuit illustration is from Refs. [41,42].

APPENDIX B: ALL POSITIVE AND NEGATIVE DONOR QAOA ANGLES PARAMETER TRANSFER ON 27- AND 16-QUBIT INSTANCES

Figure 14 shows parameter transfer performance for another set of fixed trained QAOA angles, specifically trained on the 16-qubit Ising model with only negative coefficients. Figure 15 shows the same thing but for the 16-qubit Ising model with all positive coefficients.

Figures 4, 14, and 15 all show that the different trained QAOA angles do transfer onto other problem instances. In particular, at sufficiently large p , these parameters do eventually show continuing improvement in energy as a function of p . The general finding is that despite very different random-coefficient $\{+1, -1\}$ distributions, the QAOA angles still can be effectively transferred.

APPENDIX C: LOWESA NUMERICAL SIMULATIONS

We compare PEPS simulations with a LOWESA classical numerical method, which is capable of large circuit simulations. The LOWESA simulates the expected value of a

Pauli observable by propagation in the Heisenberg picture [81,82]. The propagated observable is represented as a combination of Pauli observables. Each non-Clifford gate splits a Pauli observable into multiple Paulis, increasing computational cost. Thus, for large circuits with many non-Clifford gates, such as our QAOA circuits, a truncation scheme is necessary that discards terms with small contribution to the expected value. Here, we use a small coefficient truncation, which keeps only Pauli terms with coefficients larger than ϵ [102]. We use LOWESA implementation from the *PauliPropagation* package [102] and perform the truncation after each gate. We simulate $p = 3$ and $p = 5$ QAOA angles trained on the 16-qubit instance 0 and applied to a 127-qubit problem. We investigate the convergence of the energy and expectation values of individual Hamiltonian terms (2) versus ϵ . We perform the LOWESA simulation for each Hamiltonian term independently and use $10^{-2.25} \geq \epsilon \geq 10^{-5}$ ($10^{-2.25} \geq \epsilon \geq 10^{-4.75}$) for $p = 3$ ($p = 5$). We compare the results with the PEPS convergence with increasing bond dimension D , using $D = 2, 4, 6, 8$ [$D = 4, 8, 16, 32$ for $p = 3$ ($p = 5$)]. We gather the results in Fig. 16. We find that the LOWESA and PEPS energies converge to the same values in a limit of small

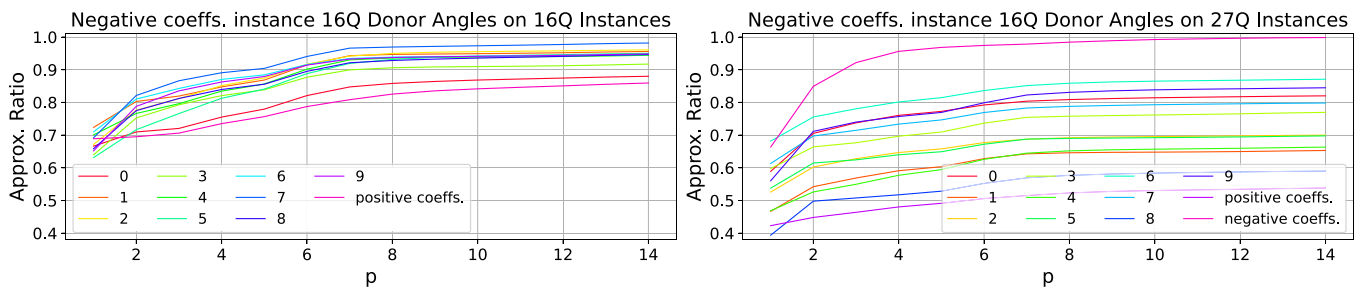


FIG. 14. Numerical simulation of parameter transfer using the donor source graph of an entirely negative coefficient Ising model, applied to the other 11 distinct 16-qubit problem instances (left) and onto the 12 separate 27-qubit problem instances (right). Notably, this set of trained QAOA angles performed well in the sense that the angles result in continuous improvement as a function of p . However, the trained angles only go up to $p = 14$ and on the 27-qubit unseen instances none of the trained QAOA angles passed an approximation ratio of 0.9 with the exception of the entirely negative-coefficient model (shown by the magenta line). Each of the ten random-coefficient models (labeled as integers in each subplot legend) correspond exactly to the same set of Ising models used in the initial QAOA parameter training, as shown in Fig. 3.

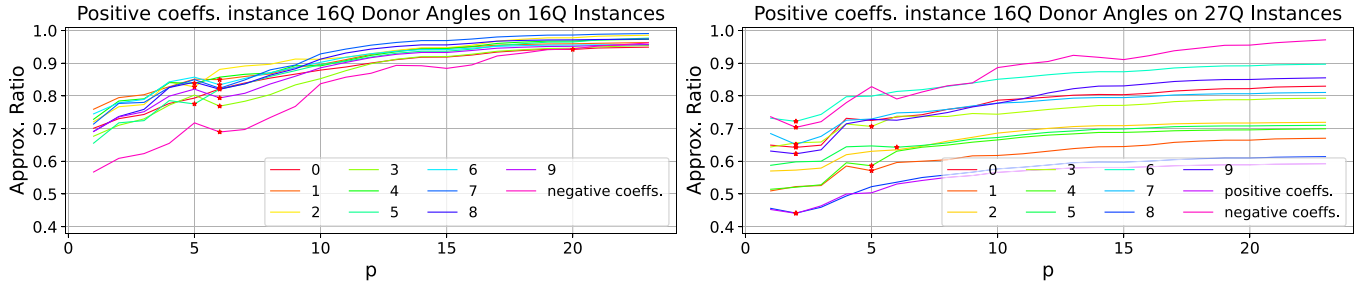


FIG. 15. Numerical simulation of QAOA parameter transfer using the donor source graph of an entirely positive coefficient model, applied to the other 11 distinct 16-qubit problem instances (left) and onto the 12 27-qubit problem instances (right). The first p step where the parameter angles have a decrease in approximation ratio compared to the previous p steps are marked with a red asterisk (if applicable). The red asterisks denote suboptimal convergence behavior of the QAOA parameters (ideally the QAOA performance should consistently improve as a function of p). Each of the ten random-coefficient models (labeled as integers in each subplot legend) correspond exactly to the same set of Ising models used in the initial QAOA parameter training, as shown in Fig. 3.

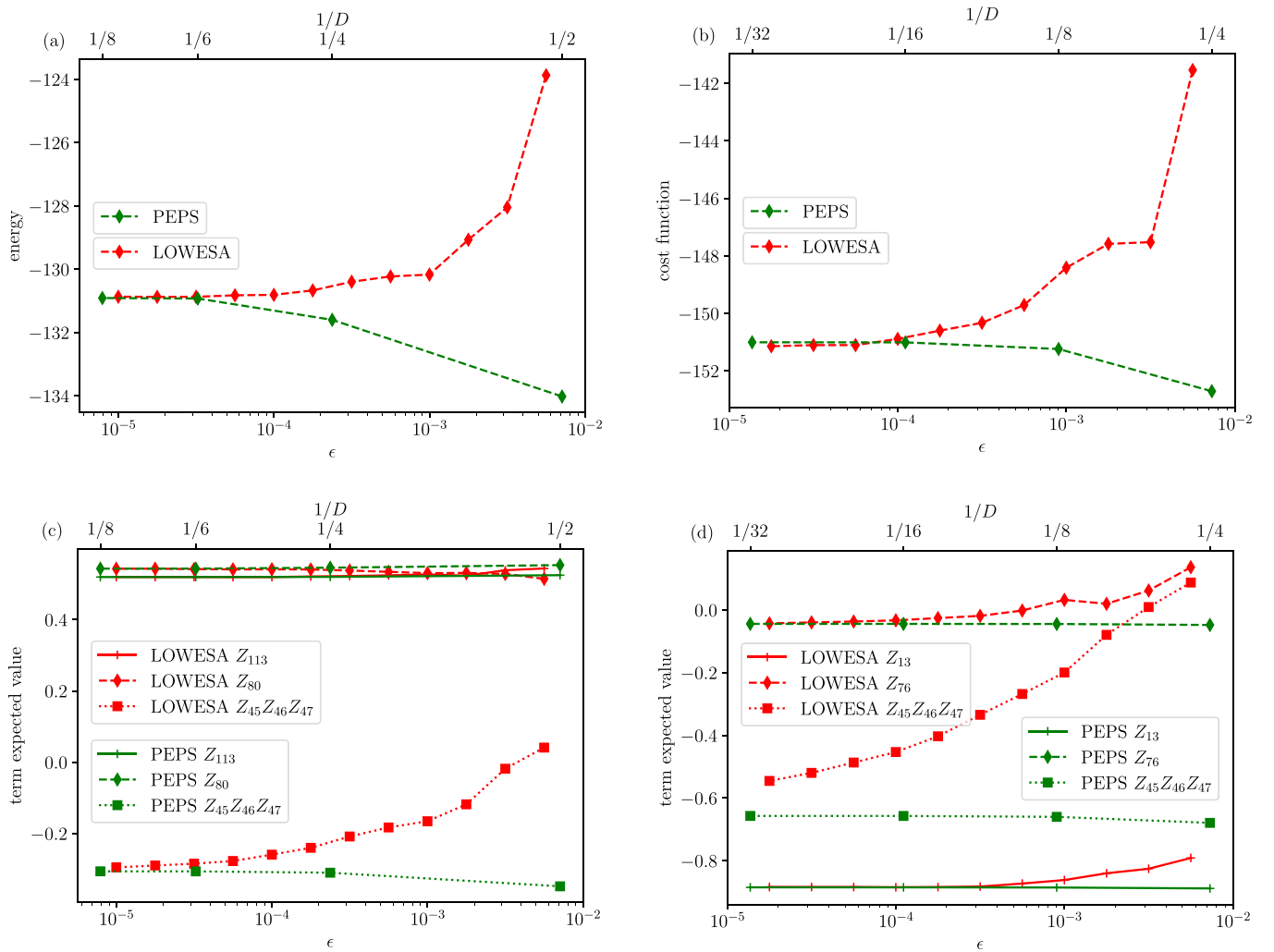


FIG. 16. A comparison of LOWESA and PEPS energy and Hamiltonian terms convergence. We plot here results for $p = 3$ [panels (a) and (c)] and $p = 5$ [panels (b) and (d)] QAOA angles trained on the 16-qubit instance 0 and applied to the 127-spin Ising model. In panels (a) and (b), we analyze convergence of the energy and in panels (c) and (d) we plot convergence of the expected values of selected Hamiltonian terms. The LOWESA results are plotted vs coefficient cutoff ϵ shown in the bottom x axis, while the PEPS simulations are plotted vs an inverse bond dimension $1/D$ in the upper axis. We use a logarithmic scale for both $1/D$ and ϵ .

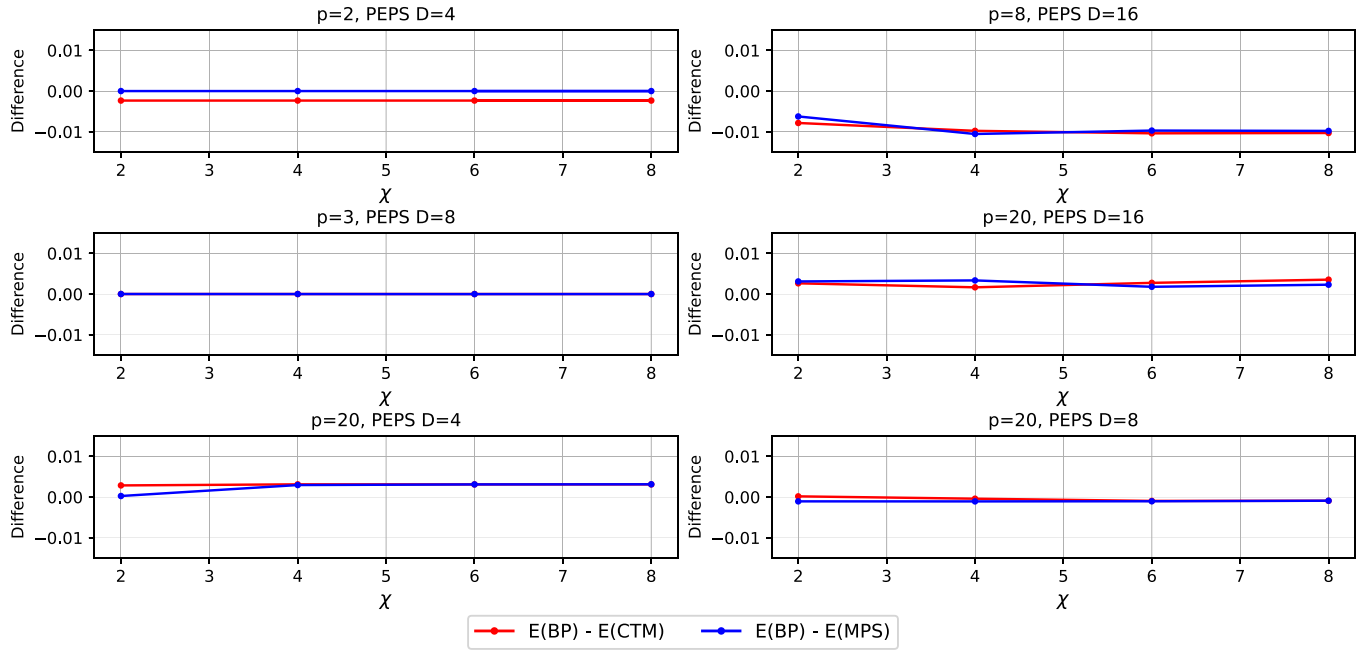


FIG. 17. Comparison of the three different PEPS contraction methods, for a few selected representative example QAOA angles, from the 16-qubit instance 0 applied to the 156-spin Ising model, at various p depths. The D in the subfigure title denotes the PEPS bond dimension. The y axes are Hamiltonian expectation value (energy, no shot noise) relative difference with respect to the expectation value computed from belief propagation. The x axes are the bond dimension χ , which here refers to the refinement parameter for the MPS and CTM methods (belief propagation has no refinement parameter, so it is constant in all plots and is therefore used to compute relative error). The y -axis scale is consistent for all subplots, showing consistently high agreement between the three approximate tensor network contraction methods.

ϵ and large D , respectively. The convergence of the Hamiltonian terms varies strongly between the terms for LOWESA. In contrast, the term expectation values converge similarly quickly in the PEPS simulations. The LOWESA estimates of the terms expected values move toward the PEPS results when ϵ is decreased, with the convergence being faster for $p = 3$. Because of this more regular convergence behavior, in main QAOA parameter transfer plots we exclusively report PEPS results. We remark that in general the LOWESA convergence depends on truncation methods and that the optimal truncation scheme for QAOA circuit simulation is an interesting open research problem.

APPENDIX D: PEPS TENSOR NETWORK CONTRACTION METHODS

We evaluate three different methods for approximate PEPS tensor network contraction: (1) BP, (2) corner transfer matrix renormalization group (CTM) [103–105] (where we adjust the procedure of Ref. [105] to the heavy-hex lattice), and (3) boundary matrix product state [106]. Both (2) and (3) have their own bond dimension, χ , which controls the precision of the approximation. Empirically, we find that BP provides both numerically stable and accurate expectation values compared to both MPS and CTM, where the contraction error is typically subleading to the finite- D PEPS representation error. This also corroborates BP-based simple update scheme for PEPS time evolution.

In Fig. 17, we show comparisons using these different approximate tensor network contraction methods. Here,

we show examples of PEPS expectation values on some subset of the QAOA angle simulations on the 156-qubit hardware-compatible Ising model instance, some of which show absolute agreement among the three tensor network contraction methods (in particular at $p = 1$) and others that show small disagreement between expectation values. In general, we see high agreement between boundary MPS, CTM, and BP methods, and therefore we use BP for all PEPS QAOA simulations reported in this study (with the exception of a handful of examples, such as in Fig. 17, where the agreement is not absolutely 0), because it is more computationally efficient than the other three methods.

APPENDIX E: REMAINING PEPS CONVERGENCE PLOTS

Figure 18 plots a complete set of PEPS convergence results for the remaining eight distinct sets of “QAOA schedules” trained on 16-qubit problem instances. Notably, the convergence properties of each set of QAOA angles, shown in the form of different source training problem instances in each subplot, vary considerably. Ranging from a lack of strong convergence at high p , such as on the $16Q-8$ instance, to very strong convergence in the case of the all-negative coefficient Ising model case (suggesting a lack of high entanglement produced by that set of QAOA circuits).

All of the convergence plots in Fig. 18, as well as Fig. 10, are normalized to use the same log-scale y axis in the interval from 9 to 10^{-6} .

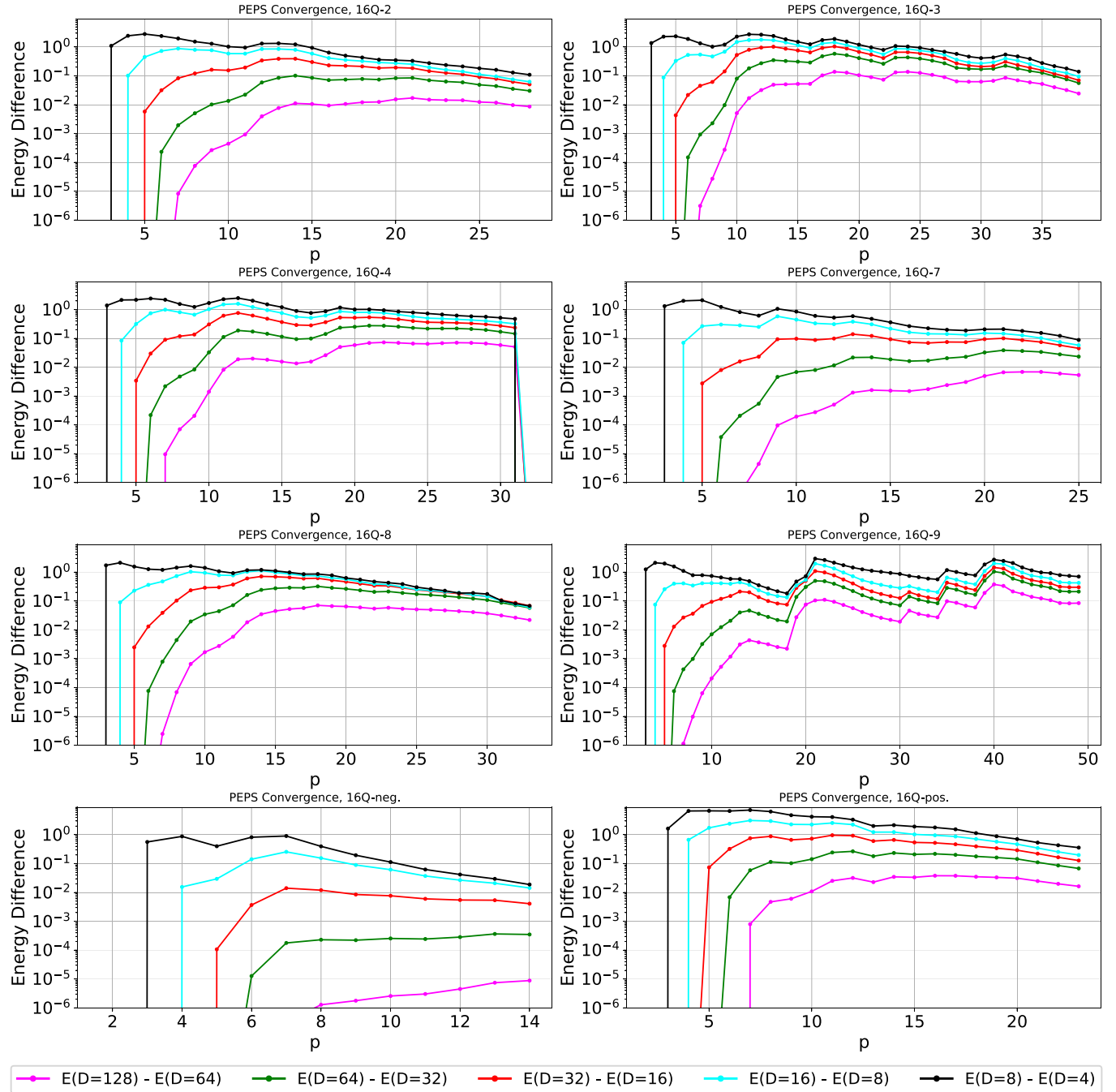


FIG. 18. PEPS convergence plots for 156-qubit QAOA circuits, using the other eight trained angle sets from 16-qubit instances, continuing from Fig. 10.

- [1] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, [arXiv:1411.4028](#).
- [2] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem, [arXiv:1412.6062](#).
- [3] S. Hadfield, Z. Wang, B. O’Gorman, E. Rieffel, D. Venturelli, and R. Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator ansatz, *Algorithms* **12**, 34 (2019).
- [4] E. Farhi, D. Gamarnik, and S. Gutmann, The quantum approximate optimization algorithm needs to see the whole graph: A typical case, [arXiv:2004.09002](#).
- [5] E. Farhi, D. Gamarnik, and S. Gutmann, The quantum approximate optimization algorithm needs to see the whole graph: Worst case examples, [arXiv:2005.08747](#).
- [6] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* **2**, 79 (2018).

- [7] E. Farhi and A. W. Harrow, Quantum supremacy through the quantum approximate optimization algorithm, [arXiv:1602.07674](#).
- [8] R. Shaydulin, *et al.*, Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem, *Sci. Adv.* **10**, eadm6761 (2024).
- [9] M. P. Harrigan, *et al.*, Quantum approximate optimization of non-planar graph problems on a planar superconducting processor, *Nat. Phys.* **17**, 332 (2021).
- [10] E. Pelofske, A. Bärttschi, J. Golden, and S. Eidenbenz, High-round QAOA for MAX k -SAT on trapped ion NISQ devices, in *IEEE International Conference on Quantum Computing and Engineering QCE'23* (IEEE, Bellevue, WA, USA, 2023), pp. 506–517.
- [11] R. Shaydulin and M. Pistoia, QAOA with $N \cdot p \geq 200$, in *IEEE International Conference on Quantum Computing and Engineering QCE'23* (IEEE, Bellevue, WA, USA, 2023), pp. 1074–1077.
- [12] J. Weidenfeller, L. C. Valor, J. Gacon, C. Tornow, L. Bello, S. Woerner, and D. J. Egger, Scaling of the quantum approximate optimization algorithm on superconducting qubit based hardware, *Quantum* **6**, 870 (2022).
- [13] J. Golden, A. Bärttschi, S. Eidenbenz, and D. O'Malley, Numerical evidence for exponential speed-up of QAOA over unstructured search for approximate constrained optimization, in *IEEE International Conference on Quantum Computing and Engineering QCE'23* (IEEE, Bellevue, WA, USA, 2023), pp. 496–505.
- [14] S. Boulebnane and A. Montanaro, Solving Boolean satisfiability problems with the quantum approximate optimization algorithm, *PRX Quantum* **5**, 030348 (2024).
- [15] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, *Nat. Rev. Phys.* **3**, 625 (2021).
- [16] The QAOA parameter similarity property comes by several different terms, including parameter transfer, problem instance independent QAOA parameters, fixed angles, parameter concentration, parameter fixing, size invariant QAOA parameters, fixed control parameters, fixed point QAOA, and transfer learning.
- [17] V. Akshay, D. Rabinovich, E. Campos, and J. Biamonte, Parameter concentration in quantum approximate optimization, *Phys. Rev. A* **104**, L010401 (2021).
- [18] X. Lee, Y. Saito, D. Cai, and N. Asai, Parameters fixing strategy for quantum approximate optimization algorithm, in *IEEE International Conference on Quantum Computing and Engineering QCE'21* (IEEE, Broomfield, CO, USA, 2021), pp. 10–16.
- [19] J. Basso, E. Farhi, K. Marwaha, B. Villalonga, and L. Zhou, The quantum approximate optimization algorithm at high depth for MaxCut on large-girth regular graphs and the Sherrington-Kirkpatrick model, in *17th Conference on the Theory of Quantum Computation, Communication and Cryptography TQC'22* (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2022), Vol. 232, pp. 7:1–7:21.
- [20] E. Farhi, J. Goldstone, S. Gutmann, and L. Zhou, The quantum approximate optimization algorithm and the Sherrington-Kirkpatrick model at infinite size, *Quantum* **6**, 759 (2022).
- [21] S. Boulebnane and A. Montanaro, Predicting parameters for the quantum approximate optimization algorithm for MAX-CUT from the infinite-size limit, [arXiv:2110.10685](#).
- [22] R. Shaydulin, P. C. Lotshaw, J. Larson, J. Ostrowski, and T. S. Humble, Parameter transfer for quantum approximate optimization of weighted MaxCut, *ACM Trans. Quantum Comput.* **4**, 1 (2023).
- [23] S. H. Sureshbabu, D. Herman, R. Shaydulin, J. Basso, S. Chakrabarti, Y. Sun, and M. Pistoia, Parameter setting in quantum approximate optimization of weighted problems, *Quantum* **8**, 1231 (2024).
- [24] J. Wurtz and D. Lykov, Fixed-angle conjectures for the quantum approximate optimization algorithm on regular MaxCut graphs, *Phys. Rev. A* **104**, 052419 (2021).
- [25] J. Wurtz and P. Love, Maxcut quantum approximate optimization algorithm performance guarantees for $p > 1$, *Phys. Rev. A* **103**, 042612 (2021).
- [26] M. Streif and M. Leib, Training the quantum approximate optimization algorithm without access to a quantum processing unit, *Quantum Sci. Technol.* **5**, 034008 (2020).
- [27] J. A. Montañez-Barrera, D. Willsch, and K. Michielsen, Transfer learning of optimal QAOA parameters in combinatorial optimization, *Quantum Inf. Process.* **24**, 129 (2025).
- [28] E. Pelofske, A. Bärttschi, L. Cincio, J. Golden, and S. Eidenbenz, Scaling whole-chip QAOA for higher-order Ising spin glass models on heavy-hex graphs, *npj Quantum Inf.* **10**, 109 (2024).
- [29] F. G. S. L. Brandao, M. Broughton, E. Farhi, S. Gutmann, and H. Neven, For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances, [arXiv:1812.04170](#).
- [30] R. Sakai, H. Matsuyama, W.-H. Tam, Y. Yamashiro, and K. Fujii, Linearly simplified QAOA parameters and transferability, [arXiv:2405.00655](#).
- [31] V. Katial, K. Smith-Miles, C. Hill, and L. Hollenberg, On the instance dependence of optimal parameters for the quantum approximate optimisation algorithm: Insights via instance space analysis, *INFORMS J. Comput.* **37**, 146 (2025).
- [32] J. Sud, S. Hadfield, E. Rieffel, N. Tubman, and T. Hogg, Parameter-setting heuristic for the quantum alternating operator ansatz, *Phys. Rev. Res.* **6**, 023171 (2024).
- [33] A. Y. Chernyavskiy, D. A. Kulikov, B. I. Bantysh, Y. I. Bogdanov, A. K. Fedorov, and E. O. Kiktenko, Improving QAOA to find approximate QUBO solutions in $O(1)$ shots, [arXiv:2509.19035](#).
- [34] A. Galda, E. Gupta, J. Falla, X. Liu, D. Lykov, Y. Alexeev, and I. Safro, Similarity-based parameter transferability in the quantum approximate optimization algorithm, *Front. Quantum Sci. Technol.* **2**, 1200975 (2023).
- [35] A. Galda, X. Liu, D. Lykov, Y. Alexeev, and I. Safro, Transferability of optimal QAOA parameters between random graphs, [arXiv:2106.07531](#).
- [36] I. Lyngfelt and L. García-Álvarez, Symmetry-informed transferability of optimal parameters in the quantum approximate optimization algorithm, *Phys. Rev. A* **111**, 022418 (2025).
- [37] A. Ozaeta, W. van Dam, and P. L. McMahon, Expectation values from the single-layer quantum approximate optimization algorithm on Ising problems, *Quantum Sci. Technol.* **7**, 045036 (2022).

- [38] Z. He, R. Shaydulin, D. Herman, C. Li, R. Raymond, S. H. Sureshbabu, and M. Pistoia, Parameter setting heuristics make the quantum approximate optimization algorithm suitable for the early fault-tolerant era, in *43rd IEEE/ACM International Conference on Computer-Aided Design ICCAD'24* (Association for Computing Machinery, New York, NY, USA, 2024), pp. 63:1–36:7.
- [39] S. H. Sack and M. Serbyn, Quantum annealing initialization of the quantum approximate optimization algorithm, *Quantum* **5**, 491 (2021).
- [40] J. A. Montañez-Barrera and K. Michielsen, Toward a linear-ramp QAOA protocol: Evidence of a scaling advantage in solving some combinatorial optimization problems, *npj Quantum Inf.* **11**, 131 (2025).
- [41] E. Pelofske, A. Bärttschi, and S. Eidenbenz, Short-depth QAOA circuits and quantum annealing on higher-order Ising models, *npj Quantum Inf.* **10**, 30 (2024).
- [42] E. Pelofske, A. Bärttschi, and S. Eidenbenz, Quantum annealing vs. QAOA: 127 qubit higher-order Ising problems on NISQ computers, in *International Conference on High Performance Computing ISC HPC'23* (Springer, Cham, 2023), pp. 240–258.
- [43] S. V. Romero, A.-M. Visuri, A. G. Cadavid, E. Solano, and N. N. Hegade, Bias-field digitized counterdiabatic quantum algorithm for higher-order binary optimization, *Commun. Phys.* **8**, 348 (2025).
- [44] S. H. Sack and D. J. Egger, Large-scale quantum approximate optimization on nonplanar graphs with machine learning noise mitigation, *Phys. Rev. Res.* **6**, 013223 (2024).
- [45] The QAOA parameter p can be found under several different terms in the literature, including *layers*, *rounds*, *depth* (importantly not the same quantity as circuit depth), and *levels*; in this study, we will typically refer to this parameter as either rounds or depth.
- [46] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, Julia: A fresh approach to numerical computing, *SIAM Rev.* **59**, 65 (2017).
- [47] J. Golden, A. Baertschi, D. O'Malley, E. Pelofske, and S. Eidenbenz, JuliQAOA: Fast, flexible QAOA simulation, in *Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis SC-W'23* (Association for Computing Machinery, New York, NY, USA, 2023), pp. 1454–1459.
- [48] J. Golden, A. Bärttschi, D. O'Malley, and S. Eidenbenz, Threshold-based quantum optimization, in *IEEE International Conference on Quantum Computing and Engineering QCE'21* (IEEE, Broomfield, CO, USA, 2021), pp. 137–147.
- [49] J. Golden, A. Bärttschi, D. O'Malley, and S. Eidenbenz, The quantum alternating operator ansatz for satisfiability problems, in *IEEE International Conference on Quantum Computing and Engineering QCE'23* (IEEE, Bellevue, WA, USA, 2023), pp. 307–312.
- [50] G. Pagano, A. Bapat, P. Becker, K. S. Collins, A. De, P. W. Hess, H. B. Kaplan, A. Kyprianidis, W. L. Tan, C. Baldwin, L. T. Brady, A. Deshpande, F. Liu, S. Jordan, A. V. Gorshkov, and C. Monroe, Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator, *Proc. Natl. Acad. Sci. USA* **117**, 25396 (2020).
- [51] S. H. Sack, R. A. Medina, R. Kueng, and M. Serbyn, Recursive greedy initialization of the quantum approximate optimization algorithm with guaranteed improvement, *Phys. Rev. A* **107**, 062404 (2023).
- [52] A. Apte, S. H. Sureshbabu, R. Shaydulin, S. Boulebnane, Z. He, D. Herman, J. Sud, and M. Pistoia, Iterative interpolation schedules for quantum approximate optimization algorithm, *arXiv:2504.01694*.
- [53] X. Lee, N. Xie, Y. Saito, D. Cai, and N. Asai, A depth-progressive initialization strategy for quantum approximate optimization algorithm, *arXiv:2209.11348*.
- [54] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, Topological and subsystem codes on low-degree graphs with flag qubits, *Phys. Rev. X* **10**, 011022 (2020).
- [55] C. Benito, E. López, B. Peropadre, and A. Bermudez, Comparative study of quantum error correction strategies for the heavy-hexagonal lattice, *Quantum* **9**, 1623 (2025).
- [56] B. Hetényi and J. R. Wootton, Creating entangled logical qubits in the heavy-hex lattice with topological codes, *PRX Quantum* **5**, 040334 (2024).
- [57] S. V. Barron, D. J. Egger, E. Pelofske, A. Bärttschi, S. Eidenbenz, M. Lehmkuehler, and S. Woerner, Provable bounds for noise-free expectation values computed from noisy samples, *Nat. Comput. Sci.* **4**, 865 (2024).
- [58] T. O'Leary, P. Czarnik, E. Pelofske, A. T. Sornborger, M. McKerns, and L. Cincio, Efficient online quantum circuit learning with no upfront training, *Commun. Phys.* **8**, 514 (2025).
- [59] E. Pelofske, A. Bärttschi, and S. Eidenbenz, Classical combinatorial optimization scaling for random Ising models on 2D heavy-hex graphs, *arXiv:2412.15572*.
- [60] A. A. Hagberg, D. A. Schult, and P. J. Swart, Exploring network structure, dynamics, and function using NetworkX, *Python in Science Conference* (2008).
- [61] A. Kotil, E. Pelofske, S. Riedmüller, D. J. Egger, S. Eidenbenz, T. Koch, and S. Woerner, Quantum approximate multi-objective optimization, *Nat. Comput. Sci.* **5**, 1168 (2025).
- [62] J. M. Chow, A. D. Córcoles, J. M. Gambetta, C. Rigetti, B. R. Johnson, J. A. Smolin, J. R. Rozen, G. A. Keefe, M. B. Rothwell, M. B. Ketchen, and M. Steffen, Simple all-microwave entangling gate for fixed-frequency superconducting qubits, *Phys. Rev. Lett.* **107**, 080502 (2011).
- [63] Qiskit contributors, Qiskit: An open-source framework for quantum computing (2023), <https://zenodo.org/record/2573505>.
- [64] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, Quantum computing with Qiskit, *arXiv:2405.08810*.
- [65] IBM ILOG CPLEX, V12.10.0: User's Manual for CPLEX, (International Business Machines Corporation, 2019), Vol. 46, p. 157, https://public.dhe.ibm.com/software/websphere/ilog/docs/optimization/cplex/ps_usrmanplex.pdf.
- [66] Gurobi Optimization, LLC, Gurobi optimizer reference manual (2024), <https://www.gurobi.com/>.
- [67] G. Vidal, Efficient classical simulation of slightly entangled quantum computations, *Phys. Rev. Lett.* **91**, 147902 (2003).
- [68] Y.-Y. Shi, L.-M. Duan, and G. Vidal, Classical simulation of quantum many-body systems with a tree tensor network, *Phys. Rev. A* **74**, 022320 (2006).

- [69] R. Jozsa, On the simulation of quantum circuits, [arXiv:quant-ph/0603163](#).
- [70] R. Alkabetz and I. Arad, Tensor networks contraction and the belief propagation algorithm, *Phys. Rev. Res.* **3**, 023073 (2021).
- [71] J. Tindall and M. Fishman, Gauging tensor networks with belief propagation, *SciPost Phys.* **15**, 222 (2023).
- [72] H. C. Jiang, Z. Y. Weng, and T. Xiang, Accurate determination of tensor network state of quantum lattice models in two dimensions, *Phys. Rev. Lett.* **101**, 090603 (2008).
- [73] J. Tindall, M. Fishman, E. M. Stoudenmire, and D. Sels, Efficient tensor network simulation of IBM's Eagle kicked Ising experiment, *PRX Quantum* **5**, 010308 (2024).
- [74] T. Begušić, J. Gray, and G. K.-L. Chan, Fast and converged classical simulations of evidence for the utility of quantum computing before fault tolerance, *Sci. Adv.* **10**, eadk4321 (2024).
- [75] S. Patra, S. S. Jahromi, S. Singh, and R. Orús, Efficient tensor network simulation of IBM's largest quantum processors, *Phys. Rev. Res.* **6**, 013326 (2024).
- [76] M. S. Rudolph and J. Tindall, Simulating and sampling from quantum circuits with 2D tensor networks, [arXiv:2507.11424](#).
- [77] M. M. Rams, G. Wójtowicz, A. Sinha, and J. Hasik, YASTN: Yet another symmetric tensor networks; a Python library for Abelian symmetric tensor network calculations, *SciPost Phys. Codebases*, **52** (2025).
- [78] M. Fishman, S. R. White, and E. M. Stoudenmire, The ITensor software library for tensor network calculations, *SciPost Phys. Codebases*, **4** (2022).
- [79] M. Fishman, S. R. White, and E. M. Stoudenmire, Codebase release 0.3 for ITensor, *SciPost Phys. Codebases*, **4-r0.3** (2022).
- [80] M. Rams, PEPS heavy-hex QAOA simulation Github, Github (2025), <https://github.com/marekrams/hex>.
- [81] E. Fontana, M. S. Rudolph, R. Duncan, I. Rungger, and C. Cirstoiu, Classical simulations of noisy variational quantum circuits, *npj Quantum Inf.* **11**, 84 (2025).
- [82] M. S. Rudolph, E. Fontana, Z. Holmes, and L. Cincio, Classical surrogate simulation of quantum systems with LOWESA, [arXiv:2308.09109](#).
- [83] R. Orús, A practical introduction to tensor networks: Matrix product states and projected entangled pair states, *Ann. Phys.* **349**, 117 (2014).
- [84] G. E. Crooks, Performance of the quantum approximate optimization algorithm on the maximum cut problem, [arXiv:1811.08419](#).
- [85] T. Müller, A. Singh, F. K. Wilhelm, and T. Bode, Limitations of quantum approximate optimization in solving generic higher-order constraint-satisfaction problems, *Phys. Rev. Res.* **7**, 023165 (2025).
- [86] P. Díez-Valle, F. J. Gómez-Ruiz, D. Porras, and J. J. García-Ripoll, Universal resources for quantum approximate optimization algorithm and quantum annealing, *Phys. Rev. Res.* **8**, 013211 (2026).
- [87] S. Boulebnane, J. Sud, R. Shaydulin, and M. Pistoia, Equivalence of quantum approximate optimization algorithm and linear-time quantum annealing for the Sherrington-Kirkpatrick model, [arXiv:2503.09563](#).
- [88] L. K. Kovalsky, F. A. Calderon-Vargas, M. D. Grace, A. B. Magann, J. B. Larsen, A. D. Baczewski, and M. Sarovar, Self-healing of Trotter error in digital adiabatic state preparation, *Phys. Rev. Lett.* **131**, 060602 (2023).
- [89] G. B. Mbeng, R. Fazio, and G. Santoro, Quantum annealing: A journey through digitalization, control, and hybrid quantum variational schemes, [arXiv:1906.08948](#).
- [90] S. Dasgupta and T. S. Humble, Characterizing the stability of NISQ devices, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Denver, CO, USA, 2020), pp. 419–429.
- [91] S. Dasgupta and T. S. Humble, Stability of noisy quantum computing devices, [arXiv:2105.09472](#).
- [92] J. Clarke and F. K. Wilhelm, Superconducting quantum bits, *Nature (London)* **453**, 1031 (2008).
- [93] E. Magesan, J. M. Gambetta, and J. Emerson, Characterizing quantum gates via randomized benchmarking, *Phys. Rev. A* **85**, 042311 (2012).
- [94] R. Harper, I. Hincks, C. Ferrie, S. T. Flammia, and J. J. Wallman, Statistical analysis of randomized benchmarking, *Phys. Rev. A* **99**, 052350 (2019).
- [95] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, Efficient Z gates for quantum computing, *Phys. Rev. A* **96**, 022330 (2017).
- [96] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, Validating quantum computers using randomized model circuits, *Phys. Rev. A* **100**, 032328 (2019).
- [97] C. H. Baldwin, K. Mayer, N. C. Brown, C. Ryan-Anderson, and D. Hayes, Re-examining the quantum volume test: Ideal distributions, compiler optimizations, confidence intervals, and scalable resource estimations, *Quantum* **6**, 707 (2022).
- [98] E. Pelofske, A. Bärttschi, and S. Eidenbenz, Quantum volume in practice: What users can expect from NISQ devices, *IEEE Trans. Quantum Eng.* **3**, 1 (2022).
- [99] P. Jurcevic, *et al.*, Demonstration of quantum volume 64 on a superconducting quantum computing system, *Quantum Sci. Technol.* **6**, 025020 (2021).
- [100] M. Dupont, N. Didier, M. J. Hodson, J. E. Moore, and M. J. Reagor, Entanglement perspective on the quantum approximate optimization algorithm, *Phys. Rev. A* **106**, 022423 (2022).
- [101] E. Pelofske, L. Cincio, M. Rams, A. Bärttschi, P. Czarnik, P. Braccia, and S. Eidenbenz, Dataset for the paper titled “Evaluating the limits of Quantum Approximate Optimization Algorithm parameter transfer at high rounds on sparse Ising models with geometrically local cubic terms”, Zenodo (2025), <http://doi.org/10.5281/zenodo.17110170>.
- [102] M. S. Rudolph, T. Jones, Y. Teng, A. Angrisani, and Z. Holmes, Pauli propagation: A computational framework for simulating quantum systems, [arXiv:2505.21606](#).
- [103] T. Nishino and K. Okunishi, A density matrix algorithm for 3D classical models, *J. Phys. Soc. Jpn.* **67**, 3066 (1998).
- [104] R. Orús and G. Vidal, Simulation of two-dimensional quantum systems on an infinite lattice revisited: Corner transfer matrix for tensor contraction, *Phys. Rev. B* **80**, 094403 (2009).
- [105] P. Corboz, S. R. White, G. Vidal, and M. Troyer, Stripes in the two-dimensional $t - J$ model with infinite projected entangled-pair states, *Phys. Rev. B* **84**, 041108(R) (2011).
- [106] F. Verstraete, V. Murg, and J. I. Cirac, Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems, *Adv. Phys.* **57**, 143 (2008).